Building Brains for Visual Traffic Analysis

Jensen, Morten Bornø

DOI (link to publication from Publisher):
10.54337/aau306590811

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Jensen, M. B. (2019). *Building Brains for Visual Traffic Analysis*. Aalborg Universitetsforlag. https://doi.org/10.54337/aau306590811

BUILDING BRAINS FOR VISUAL TRAFFIC ANALYSIS

BY
MORTEN BORNØ JENSEN

DISSERTATION SUBMITTED 2018

AALBORG UNIVERSITY
DENMARK

Building Brains for Visual Traffic Analysis

Ph.D. Dissertation

Morten Bornø Jensen

Dissertation submitted December 21, 2018

Dissertation submitted:	December 21, 2018
PhD supervisor:	Professor Thomas B. Moeslund Aalborg University
PhD committee:	Associate Professor Claus Brøndgaard Madsen (chair.) Aalborg University
	Associate Professor Brendan Morris University of Nevada
	Senior Director of Computer Vision Anthony Hoogs Kitware
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-372-3

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2019

Curriculum Vitae

Morten Bornø Jensen

Morten Bornø Jensen received his Bachelor's degree in Computer Engineering on the topic of Information Processing Systems in 2013 and his MSc in Vision, Graphics and Interactive Systems in 2015, both from Aalborg University, Denmark. From July 2012 to October 2014 Morten was employed as Student Software Developer at Intel Mobile Communications Aps in Aalborg, Denmark, main tasks concerned developing software to RF drivers. In May 2016, he embarked on the Ph.D. study with the Visual Analysis of People Lab at the section of Media Technology, Aalborg University.

Furthermore, Morten is an alumnus of the Laboratory for Intelligent and Safe Automobiles and the Computer Vision and Robotics Research Laboratory, both at the University of California, San Diego

His main interests are computer vision and machine learning, especially in the area of detecting vulnerable road users and traffic lights. He has been involved in the supervision of undergraduate and graduate students within image processing and computer vision.

Abstract

Making a computer detect and understand traffic objects automatically - be it vehicles, pedestrians or traffic lights - are essential components for self-driving cars or automated traffic analysis. This Ph.D. thesis addresses four computer vision topics in respect to traffic analysis, which are Video Acquisition, Object Detection, Semantic, and Knowledge for the World.

In Video Acquisition, a multi-modal and multi-view recording setup is developed which has been used for capturing several months of video data in Europe. To aid less occlusion-prone capturing view-angles, we provide an overview of available portable poles, which shows a lack of a lightweight yet robust portable pole. To this end, we develop one satisfying these requirements. Finally, we have assembled a traffic intersection dataset with synced view-angles from both a drone and existing infrastructure. Preliminary results using Mask R-CNN show that the drone view-angle is superior when detecting vehicles but inferior when detecting pedestrians and cyclists.

Within Object Detection, a traffic light dataset has been assembled and published together with an extensive survey of state of the art within traffic light recognition. The survey indicated that detecting traffic lights had not followed the same machine learning advancement as in similar detection areas. To this end, we by brought the traffic light detection up to par using the Aggregate Channel Features detector and the You-Only-Look-Once detector.

For Semantic, we propose a human-in-the-loop traffic analysis watchdog tool for non-practitioners of computer vision. The tool is based on traditional computer vision methods and is made very user-friendly causing it to be used worldwide. Further, an automatic data-driven traffic analysis tool based on the Single Shot MultiBox Detector is proposed. The tool produces object trajectories in world coordinates enabling further conflict analysis or the likes.

In the last topic, Knowledge for the World, a non-peer-reviewed popular science paper on Deep Learning has been published targeted towards high school student, spawning several inquiries on the perspectives and how to get started. Further, we attended the Danish Folkemøde 2018, "The People's Meeting." We showcased the latest advancement within computer vision which led to two live TV interviews.

Resumé

At få en computer til at finde og forstå trafikobjekter automatisk - for eksempel køretøjer, fodgængere eller trafiklys - er vigtige komponenter for selvkørende biler eller automatiseret trafikanalyser. Denne Ph.D. afhandling omhandler fire computer vision emner i forhold til trafik: Video Anskaffelse, Objektdetektion, Semantik og Viden for Verden.

I Video Anskaffelse er der udviklet et multi-modalt og multi-view optage opsætning, som er blevet brugt til at indsamle flere måneders videodata i Europa. For at forbedre synsvinklen for video anskaffelsen, giver vi et overblik over tilgængelige mobile master, hvilket viser manglen på en let, men robust mobil mast. Derfor udvikler vi en, der opfylder disse krav. Endelig har vi samlet et trafikkrydsdatasæt med synkroniseret data fra både en drone og eksisterende infrastruktur. Preliminære resultater ved hjælp af Mask R-CNN viser, at drone-synsvinklen er overlegen, når det registrerer køretøjer, men dårligere, når man registrerer fodgængere og cyklister.

Inden for Objektdetektion er et trafiklysdatasæt blevet samlet og udgivet sammen med en omfattende undersøgelse af state of the art inden for trafiklys genkendelse. Undersøgelsen viste, at detekteringen af trafiklys ikke har fulgt samme maskinindlæringsfremskridt som i lignende detekteringensområder. Derfor har vi bragt disse fremskridt til trafiklysdetekteringen ved hjælp af Aggregate Channel Features metoden og You-Only-Look-Once metoden.

For Semantik foreslår vi et menneske-i-loop-trafikanalyse værktøj til ikke-udøvere af computer vision. Værktøjet er baseret på traditionelle computer vision metoder og er lavet meget brugervenligt, hvilket har medført brug i hele verden. Endvidere foreslås et automatisk data-drevet trafikanalyse værktøj baseret på Single Shot MultiBox detektor. Værktøjet producerer trajektorier i verdenskoordinater, der muliggør yderligere konfliktanalyse el.lign.

I det sidste emne, Viden for Verden, er en ikke-fagfællesbedømt populærvidenskablig artikel om Deep Learning udgivet rettet mod gymnasieelever. Dette har affødt flere henvendelser om perspektiverne og hvordan man kommer i gang. Derudover deltog vi ved Folkemøde 2018. Her fremviste vi den seneste udvikling inden for computer vision, hvilket førte til to live tv-interviews.

Contents

Thesis Details

Thesis Title: Building Brains for Visual Traffic Analysis
Ph.D. Student: Morten Bornø Jensen
Supervisor: Professor Thomas B. Moeslund, Aalborg University

The main body of this thesis consists of the following papers:

Video Acquisition

[A] Morten B. Jensen and Thomas B. Moeslund, "InDeV Recording Setup." In: *InDeV Technical Report*, Nov. 2016 (revised in Dec. 2018).

[B] Morten B. Jensen, Chris H. Bahnsen, Harry Lahrmann, Tanja K. O. Madsen, and Thomas B. Moeslund, "Collecting Traffic Video Data using Portable Poles: Survey, proposal, and analysis." In: *Journal of Transportation Technologies*. SCIRP, Vol. 8 No. 4, pp. 376–400, 2018.

[C] Morten B. Jensen, and Thomas B. Moeslund, "Multi-view Traffic Intersection Dataset: Performance Analysis and Comparison." *Ongoing*, 2018.

Object Detection

[D] Morten B. Jensen, Mark P. Philipsen, Andreas Møgelmose, Thomas B. Moeslund, and Mohan M. Trivedi, "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives." In: *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800-1815, 2016.

[E] Morten B. Jensen, Mark P. Philipsen, Thomas B. Moeslund, and Mohan M. Trivedi, "Comprehensive Parameter Sweep for Learning-Based Detector on Traffic Lights." In: *Advances in Visual Computing: 12th International Symposium on Visual Computing, ISVC 2016*. Springer. Lecture Notes in Computer Science vol. 10073, pp. 92–100, 2016.

[F] Morten B. Jensen, Kamal Nasrollahi, and Thomas B. Moeslund, "Evaluating State-of-the-art Object Detector on Challenging Traffic Light Data." In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW): Traffic Surveillance Workshop and Challenge*, pp. 882-888, 2017.

[G] Morten B. Jensen, Mark P. Philipsen, Chris Bahnsen, Andreas Møgelmose, Thomas B. Moeslund, and Mohan M. Trivedi, "Traffic Light Detection at Night: Comparison of a Learning-based Detector and three Model-based Detectors." In: *Advances in Visual Computing: 11th International Symposium on Visual Computing, ISVC 2015*. Springer. Lecture Notes in Computer Science vol. 9474, pp. 774–783, 2015.

[H] Martin Ahrnbom, Morten B. Jensen, Kalle Åström, Mikael Nilsson, Håkan Ardö, and Thomas B. Moeslund, "Improving a real-time object detector with compact temporal information." In: *IEEE International Conference on Computer Vision Workshop (ICCVW): Computer Vision for Road Scene Understanding and Autonomous Driving workshop*, pp. 190–197, 2017.

Semantic

[I] Chris H. Bahsen, Tanja K. O. Madsen, Morten B. Jensen, Harry S. Lahrmann, and Thomas B. Moeslund, *The RUBA Watchdog Video Analysis Tool*, Deliverable submitted within the InDeV EU project, 2018, available at `https://www.indev-project.eu/InDeV/EN/Documents/pdf/4-3.pdf?__blob=publicationFile&v=2`. The deliverable is based on the public wiki page at `https://bitbucket.org/aauvap/ruba/wiki/`.

[J] Morten B. Jensen, Martin Ahrnbom, Maarten Kruithof, Kalle Åström, Mikael Nilsson, Håkan Ardö, Aliaksei Laureshyn, Carl Johnsson, and Thomas B. Moeslund, "A Framework For Automated Analysis of Surrogate Measures of Safety from Video using Deep Learning Techniques." In: *Transportation Research Board (TRB) 98th Annual Meeting*. In Press, 2019.

Knowledge for the World

[K] Morten B. Jensen, Chris H. Bahnsen, Kamal Nasrollahi, and Thomas B. Moeslund, "Deep Learning - et gennembrud indenfor kunstig intelligens." In *Aktuel Naturvidenskab*, Vol. 2, pp. 8–13, 2018.

[L] Morten B. Jensen, Malte Pedersen, and Poul Lund, "Techtunnel - Folkemødet 2018." In: *AAU Tech report*, 2018.

In addition to the main papers listed above, the following publications have been co-authored in connection with the Ph.D.:

- Morten B. Jensen, Mark P. Philipsen, Mohan M. Trivedi, Andreas Møgelmose, and Thomas B. Moeslund, "Ongoing Work on Traffic Lights: Detection and Evaluation." In: *IEEE International Conference on Advanced Video and Signal-based Surveillance*, pp. 1–6, 2015.

- Morten B. Jensen, Rikke Gade, and Thomas B. Moeslund, "Swimming Pool Occupancy Analysis using Deep Learning on Low Quality Video." In: *Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports*. ACM. pp. 67–73, 2018.

- Mark P. Philipsen, Morten B. Jensen, Ravi K. Satzoda, Mohan M. Trivedi, Andreas Møgelmose, and Thomas B. Moeslund, "Day and night-time drive analysis using stereo vision for naturalistic driving studies." In *IEEE Intelligent Vehicles Symposium (IV), Proceedings*, pp. 1226–1231, 2015.

- Mark P. Philipsen, Morten B. Jensen, Andreas Møgelmose, Thomas B. Moeslund, and Mohan M. Trivedi, "Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset." In: *IEEE International Conference on Intelligent Transportation Systems*, pp. 2341–2345, 2015.

- Jesper B. Pedersen, Jonas B. Markussen, Mark P. Philipsen, Morten B. Jensen, and Thomas B. Moeslund, "Counting the Crowd at a Carnival." In: *Advances in Visual Computing: 10th International Symposium on Visual Computing, ISVC 2014*. Springer. Lecture Notes in Computer Science vol. 8888, pp. 706–715, 2014.

- Huamin Ren, Hong Pan, Søren I. Olsen, Morten B. Jensen, and Thomas B. Moeslund, "An In-depth Study of Sparse Codes on Abnormality Detection." In: *IEEE Advanced Video and Signal-based Surveillance (AVSS)*, pp. 1–7, 2016.

- Tanja K. O. Madsen, Peter M. Christensen, Chris H. Bahnsen, Morten B. Jensen, Thomas B. Moeslund, and Harry S. Lahrmann, "RUBA-Videoanalyseprogram til trafikanalyser." In: *Trafik og Veje*, Vol. 3, pp. 14–17, 2016.

Further, this thesis has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895 with the project name: "In-Depth understanding of accident causation for Vulnerable road users" (InDeV). In addition to publication I, this has resulted in the following publications:

- Tanja K. O. Madsen, Niels Agerholm, András Várhelyi, Chris H. Bahnsen, Morten B. Jensen, Aliaksei Laureshyn, Thomas B. Moeslund, and Harry S. Lahrmann, "Tools for Naturalistic VRU Study-Hands-on Manual" in *European Commission Office for Official Publications of the European Union*, 2018.

- Tanja K. O. Madsen, Camilla S. Andersen, András Várhelyi, Mikael Nilsson, Magnus Oskarsson, Morten B. Jensen, Chris H. Bahnsen, Mads B. Christensen, and Thomas B. Moeslund, "Mobile Application for Naturalistic Walking/Cycling Data Collection" in *European Commission Office for Official Publications of the European Union*, 2017.

- Tanja K. O. Madsen, Chris H. Bahnsen, Morten B. Jensen, Harry S. Lahrmann, and Thomas B. Moeslund, "Watchdog System" in *European Commission Office for Official Publications of the European Union*, 2016.

The thesis is based on ongoing research and published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This thesis is submitted as a collection of papers in fulfillment of a Ph.D. study at the Section of Media Technology, Aalborg University, Denmark. The work consists of research on four main topics: Video Acquisition, Object Detection, Semantic, and Knowledge for the World. Thus, this thesis is organized with five parts. Part one contains an introduction and overview of the state-of-the-art in the four main topics as well as the contributions which have been made to them during this work. This is followed by a part containing selected papers published during this Ph.D. in each of the four topics mentioned above.

This project has been carried out from 2015-2018, mainly in the Visual Analysis of People Lab (VAP) at Aalborg University, but with one research stay in the Laboratory of Intelligent and Safe Automobiles at University of California, San Diego.

In truth, I could not have achieved this level of success without strong mentors, encouraging friends, and my loving family. I thus first and foremost want to thank my mentor and supervisor, Prof. Thomas B. Moeslund, for his excellent guidance throughout both my Master's and Ph.D. He has given me all the freedom and encouragement needed to pursue my research interests while ensuring that I stayed on track. Further, I am thankful for having had the privilege of working with Prof. Mohan M. Trivedi, who always showed a profound interest in my research while providing enthusiastic guidance.

An enormous thanks to all my current and previous colleagues in VAP, in particular, Mark P. Philipsen, who I started this academic adventure with and has since enjoyed many great moments with. I would also like to thank my friends Aksel Bang, Jesper B. Pedersen, Jonas B. Markussen, and Mikkel K. Hyttel for encouraging me when I felt hopeless.

Finally, I would like to dedicate this work to my always supporting parents, my inspiring brother, and my loving grandparents; and thank them all for instilling in me, the desire of learning to learn.

Morten Bornø Jensen
Aalborg University, December 21, 2018

Part I

Overview of the work

Chapter 1

Introduction

Humankind has from our very origin been innovative. We have been drawn towards inventions and discoveries that aid our survival and adaption to an ever developing and changing the world - each new invention and discovery leading on from the previously discovered ones. Through millenniums of new inventions and discoveries, the foundation of the industrial age and industrial revolution were around the mid-18th century [1]. The revolution itself can be characterized as the period where the hard manual labor was aided or even, in some areas, automated by the use of machinery, e.g., steam engines [2, 3]. The reason for adopting and implementing automation can be many and very different, for the British people during the industrial revolution, the material living standards improved [1], which also lead to an economic boost. Today, the scope of automation is still often related to the economic reward, as the purpose is mostly to improve the profit margin of a given product while keeping the return of automation investment as short as possible. But automating a given task can also provide a more consistent quality of your product, which in some industries are vital and essential. The goals and purpose can vary a lot, but typical for them all is that they today are often carried out by the implementation of an artificial intelligence combined with mechanical robots and sensors [4–6].

Artificial intelligence is quite a 'buzz words' at the time of writing, but that does not necessarily mean that it does not have some hang to it. Automation tasks using modern technology are heavily based on an artificial intelligence, e.g., a computer, which comes in many shapes. The computer is, by the use of one or more sensors, able to analyze its surrounding which is an essential and key component in making the computer able to understand the world and thus act upon it, e.g., by the use of mechanical devices such as robots. This field of making the computer intelligent in the sense that it analysis and understands its surrounding is called computer vision. Throughout the last 50 years, computer vision has grown to become a well established research area, which have proven its benefits in many vastly different fields such as agriculture [7], the world of sports [8], quality control in automated manufacturing systems [9], transportation [10], and so many more.

Especially the transportation field has been quite hyped due to the autonomous vehicles and driving assistant systems (DAS) [10, 11] such as pedestrian detection, lane detection, and traffic light detection. All of these applications do heavily rely on computer vision. In the same way, computer vision applications is spawned in traffic surveillance where monotonously and manual tasks such as traffic flow estimations, vehicles counts, and traffic intersection monitoring, are being replaced by automatic computer vision systems that can do more consistent traffic flow estimations, abnormality detections, traffic intersection monitoring for safety analysis, etc. The field of transportation persist a facinating and challenging field as studies show that people in the age group of 20-59 years spend more than one hour daily traveling in automobiles [12], which unfortunately also result in accidents and near-accidents. In the European Union (EU) alone, more than 25,000 people died, and approximately 135,000 people were seriously injured as a result of accidents across the roads in 2017 [13]. This of course is first and foremost an incredible personal lose for the involved families, but as a society, it is also an enormous economic loss.

As a result, the EU set out a 2010-2020 goal with an overall objective of halving road deaths across Europe. Several initiatives have been started within the Framework HORIZON2020 by the European Commission [14]. One of the efforts is the "In-depth Understanding of Accident Causation for Vulnerable Road Users" (InDev) research project [15], which partly founds this thesis. One of the main research topics within this Ph.D. thesis is to promote the use of modern technology, i.e., computer vision, to increase road safety.

Whether a computer vision system is detecting traffic lights or Vulnerable Road Users (VRUs), a general high-level concept applies to most computer vision systems, which is illustrated in the block diagram in Figure 1.2. This block diagram is a very general representation of the stages used when developing a computer vision system.

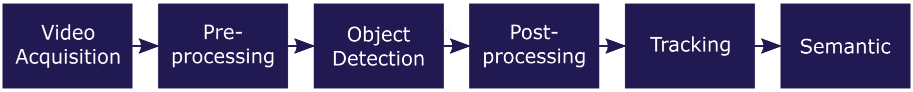


Fig. 1.1: Overview of a general computer vision pipeline with six computer vision stages.

The starting point of most computer vision systems is Video Acquisition, which mostly concerns sensor choice, which is not necessarily limited to the popular RGB camera. Further, an important thing is how the video acquisition setup is created and carried out in order to get useful data with respect to the given application. Pre-processing is the first stage that allows for modification of the acquired data, which may need some adjustment, e.g., masking, but it could also involve manipulating the image to extract proper features. The features are used in the Object Detection stage, where the title itself reveals the scope. After detecting a set of objects in your data, we can do some cleaning of the detections in post-processing, where after the detections can be connected in spatiotemporal space, meaning that each detected object in the video needs to be either associated with a previously existing track or as an entirely new track in the video [16]. Finally, we can use the produced trajectories of the objects to generate some semantic interpretation and understanding on a more high-level, e.g., behavioral analysis [17].

FOCUS OF THIS THESIS

This thesis tackles the stages marked with dark blue in Figure 1.2: Video Acquisition, Object Detection, and Semantic, with respect to mainly traffic light detection from a DAS perspective and automated traffic analysis from a traffic surveillance perspective. These three stages are hereafter described as main topics.

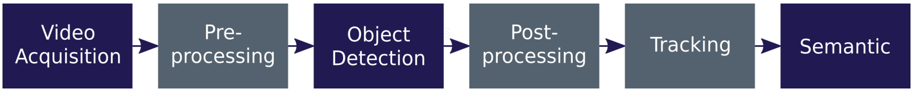


Fig. 1.2: The computer vision stages involved in this thesis are marked in dark blue. The stages that will not be covered are grayed out.

Further, a focus point of this thesis has been to disseminate and translate the use of modern technology, i.e., computer vision, to the public, which is titled as per the Aalborg University slogan and strategy: Knowledge for the World. Together with the aforementioned main topics, these four main topics will constitute the outline and focus of this thesis and are introduced in the following sections.

1 VIDEO ACQUISITION

Video Acquisition is the first stage in the overall computer vision concept, as illustrated in Figure 1.2. Acquiring video data can be done with a large set of sensors, of which the most well-known and wildly used sensor is the RGB camera. In fact, most of us carry an RGB camera in our pockets as it is the camera installed in most of our cell phones. In traffic surveillance, the thermal camera has also seen a usage boost during the last decade as it provides data that is more robust to weather and lighting conditions as illustrated in Figure 1.3.

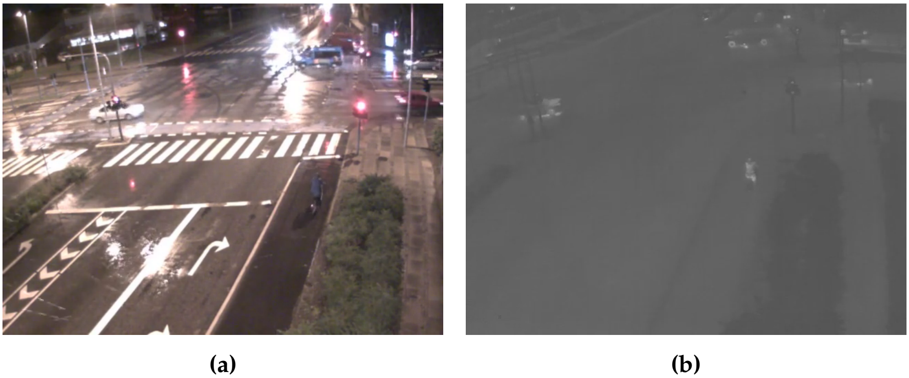


Fig. 1.3: Example of the (a) RGB modality and (b) thermal modality captured at a traffic intersection simultaneously during a rainy night. Notice the strong reflections in the RGB image, as well as the poor contrast in the thermal image. *Image source: [16]*

However, the data used for the remaining of our computer vision pipeline do not necessarily need to be of video, it could also be point clouds from time-of-flight sensors.

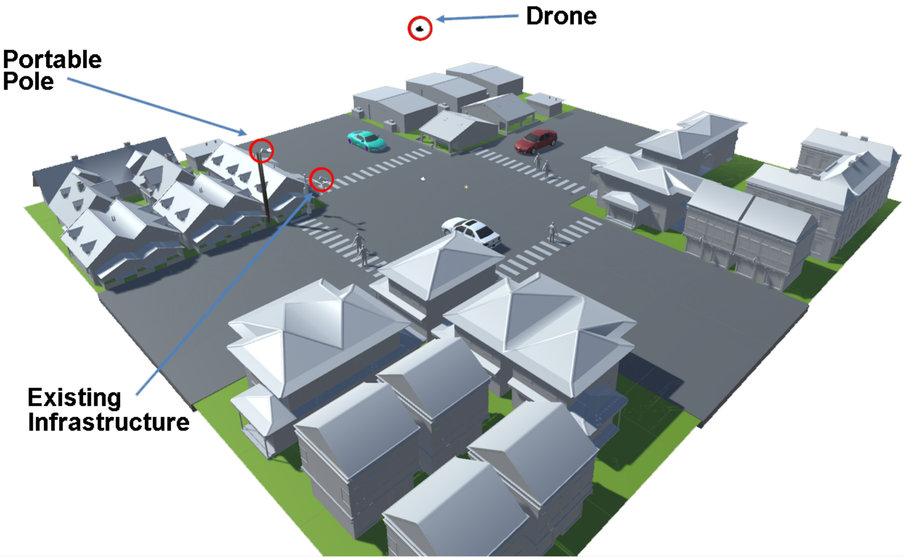


Fig. 1.4: A traffic intersection equipped with three different capturing view-angels. The three options are by mounting camera equipment in existing infrastructure, a portable pole, and utilizing a drone equipped with a camera. *Image source: [16]*

Obviously, it is essential to select a proper sensor or sensors with respect to the problem the computer vision system aims to solve. However, for large-scale studies such as the one addressed in the InDeV project, selecting the correct sensor is only a part of solving the problem. Defining and developing the proper recording setup is just as important as the specific sensor choice, such the acquired data is correct and usable.

In recent years the usage of drones has become quite popular for traffic surveillance and traffic analysis as it provides a top-down bird view-angle as illustrated in Figure 1.4. Traditionally, data have been collected by the use of installing camera equipment existing infrastructure or by the use of a portable pole, which is illustrated in Figure 1.4.

This thesis present works related to capturing video data in large-scale studies. Further, it investigates the pros and cons of capturing data from existing infrastructure, portable poles, and from drones with respect to performing automated traffic analysis using computer vision.

2 OBJECT DETECTION

The goal of this main topic is to localize objects which in the perspective of general computer vision can vary a lot. Detecting objects in the acquired video is often the key component in the computer vision system. Object detection is essential in autonomous vehicles, where the surrounding objects such as pedestrians, vehicles, signs, traffic lights, traffic lanes, etc. are of particular interest as this provide the vehicle with information how it can safely navigate given its surroundings. Another example of object detection playing a key role is in in a manufacturing line where the computer vision system detects errors and flaws as part of the quality control of a given product.

In layman's terms, well working object detection is often characterized as well working computer vision, as it is usually the main reason to apply computer vision in the first place. The outcome of object detection can be many and is of course very application depended, but general for them all is that it involves locating the object or objects in the image, e.g., as axis-aligned bounding boxes with a corresponding class label, i.e., if you looking for multiple object classes, as seen in Figure 1.5.

This thesis presents work related to detection of traffic lights from a DAS perspective as well as detection of objects relevant for automated traffic analysis.

3 SEMANTIC

Though computer vision in itself is a broad research field, it also serves as a tool for a lot of other research fields and applications. For non-practitioners

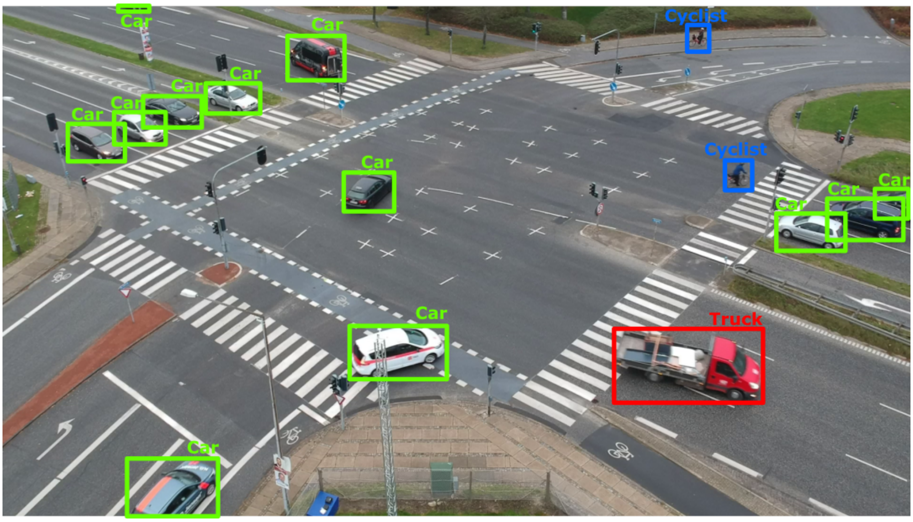


Fig. 1.5: Detecting cars, cyclists, and trucks are an essential part of automated traffic analysis. *Image source: [16]*

of computer vision, it is not necessarily the scope to develop state-of-the-art computer vision algorithms; they are in most cases only interested in how computer vision can benefit and automate tedious tasks within their research field.

In order to make the low-level information, e.g., digital representations of features or objects in the data, extracted using the computer vision system, perceivable and usable for non-practitioners of computer vision; the extracted information needs to be translated from the low-level information into high-level semantic information, e.g., drive behavior, observed actions, and event recognition. Semantics are very much a mathematical logic that can be triggered by a pre-defined combination of events which are very application dependent. However, given an event that can be mathematically defined in respect to a logic in the extracted low-level information, large quantities of data can, by the use of a computer vision system with focus on high-level semantic understanding, be reduced to a pre-defined set of perceivable metrics.

This thesis presents work on how to provide high-level semantic information in respect to automated traffic analysis of large-scale studies.

4 KNOWLEDGE FOR THE WORLD

The overall strategy for Aalborg University (AAU) focuses on Knowledge for the World, which is implemented in all the core activities within the University, i.e., education, research, and knowledge collaboration. Since AAU was established in 1974 one of the key goals has been to be and remain a vital platform for generating global knowledge across many disciplines, which ultimately aids, develop, and makes a difference in the local communities as well as the entire society [18].

This main topic does not in its current form constitute proper peer-reviewed research; it serves an entirely different purpose. Being a Ph.D. student and researcher at AAU obligates and challenges one to make a difference for the public society by collaborating with external and multi-disciplinary partners while disseminating the advancements done within the researchers field of study to the public.

This thesis present initiatives carried out to involve the public and improve the knowledge of artificial intelligence and computer vision for the common man, in particular, our younger generation.

5 THESIS STRUCTURE

This thesis is divided into five parts. Part I contains an introduction to each of the four main topics: Video Acquisition, Object Detection, Semantic, and Knowledge for the World. For each of the four main topics in Part I, an introduction and a general overview of the state-of-the-art for the specific topic will be given along with the contributions made from this Ph.D. work. Finally, a general and overall conclusion is made, completing Part I.

Part I serves as a general introduction and overview covering all of the Ph.D. work and will thus not delve deeply into the details of the work carried out during this Ph.D.. As this thesis is submitted and structured as a collection of papers, the general introduction in Part I is based upon the enclosed articles allowing the reader to delve into the details if desired. After the introduction and overview in Part I, Part II through V contains the included papers for each of the four aforementioned main topics. Besides the included papers submitted with this thesis, additional papers have been written during the Ph.D., and while they are not included in this thesis, they will be referenced whenever appropriate. Finally, each chapter contains its own bibliography.

REFERENCES

[1] T. S. Ashton, *The Industrial Revolution 1760-1830*, ser. OUP Catalogue. Oxford University Press, 1997, no. 9780192892898. [Online]. Available: https://ideas.repec.org/b/oxp/obooks/9780192892898.html

[2] T. Sheridan, T. Vámos, and S. Aida, "Adapting automation to man, culture and society," *Automatica*, vol. 19, no. 6, pp. 605 – 612, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109883900249

[3] R. C. Allen *et al.*, *The British industrial revolution in global perspective.* Cambridge University Press Cambridge, 2009, vol. 1.

[4] C. Dirican, "The impacts of robotics, artificial intelligence on business and economics," *Procedia-Social and Behavioral Sciences*, vol. 195, pp. 564–573, 2015.

[5] C. S. Tirgul and M. R. Naik, "Artificial intelligence and robotics," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 5, no. 6, pp. 1787–1793, 2016.

[6] S. A. Wright and A. E. Schultz, "The rising tide of artificial intelligence and business automation: Developing an ethical framework," *Business Horizons*, vol. 61, no. 6, pp. 823 – 832, 2018, eTHICS, CULTURE, AND PEDAGOGICAL PRACTICES IN THE GLOBAL CONTEXT. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007681318301046

[7] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Computers and Electronics in Agriculture*, vol. 153, pp. 69 – 81, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169918305829

[8] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, "Computer vision for sports: Current applications and research topics," *Computer Vision and Image Understanding*, vol. 159, pp. 3 – 18, 2017, computer Vision in Sports. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314217300711

[9] E. Asoudegi and Z. Pan, "Computer vision for quality control in automated manufacturing systems," *Computers & Industrial Engineering*, vol. 21, no. 1, pp. 141 – 145, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/036083529190078K

[10] R. P. Loce, R. Bala, and M. Trivedi, *Computer Vision and Imaging in Intelligent Transportation Systems.* John Wiley & Sons, 2017.

[11] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *Trans. Intell. Transport. Sys.*, vol. 8, no. 1, pp. 108–120, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1109/TITS.2006.889442

[12] N. C. McDonald, “Trends in automobile travel, motor vehicle fatalities, and physical activity: 2003-2015,” *American Journal of Preventive Medicine*, vol. 52, no. 5, pp. 598 – 605, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0749379716306936

[13] European Commission, *2017 road safety statistics: What is behind the figures? - Fact Sheet*. European Commission, 2018. [Online]. Available: http://europa.eu/rapid/press-release_MEMO-18-2762_en.pdf

[14] European Commission, *Towards a European Road Safety Area: Policy Orientations on Road Safety 2011-2020*. European Commission, 2010. [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/road_safety/pdf/com_20072010_en.pdf

[15] E. Polders and T. Brijs, “How to analyse accident causation? a handbook with focus on vulnarable road users,” 2018.

[16] M. Jensen, M. Ahrnbom, M. Kruithof, K. Åström, M. Nilsson, H. Ardö, A. Laureshyn, C. Johnsson, and T. Moeslund, “A framework for automated analysis of surrogate measures of safety from video using deep learning techniques.” Transportation Research Board National Cooperative Highway Research Program, 9 2018.

[17] B. T. Morris, M. M. Trivedi *et al.*, “A survey of vision-based trajectory learning and analysis for surveillance,” *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 8, p. 1114, 2008.

[18] Aalborg University, *Knowledge for the World - Aalborg University Strategy 2016-2021*. Aalborg University, 2015. [Online]. Available: http://www.e-pages.dk/aalborguniversitet/383/

Chapter 2

Video Acquisition

The entire computer vision pipeline is linked resulting in each stage being depended on the quality and result of the previous stage, which inevitably causes the briefly mentioning and perspectives to the following stage in the computer vision pipeline.

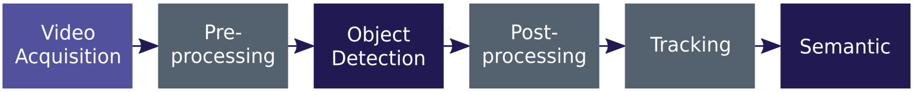


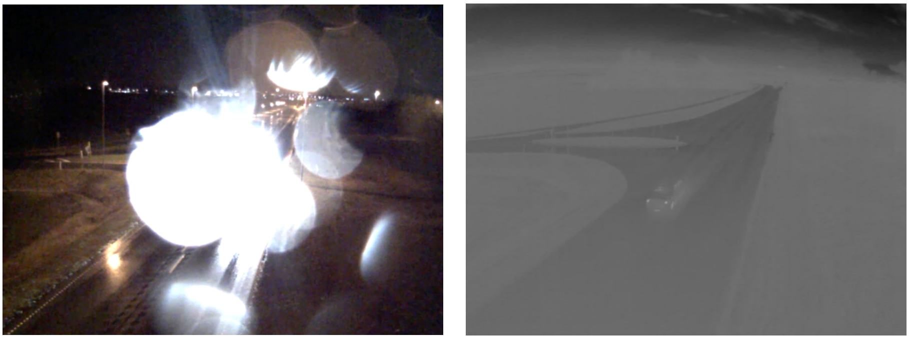
Fig. 2.1: Video Acquisition is the first stage in the overall computer vision pipeline and is thus vital for the remaining of the computer vision system.

Video Acquisition is a vast and application depended topic but is in all cases a necessity as it provides input to remaining of the pipeline as seen in Figure 2.1. Though the core considerations can to some extent be similar across the various applications, this chapter will take its starting point in Video Acquisition with the perspective of doing large-scale traffic analysis studies at critical traffic intersection across varying lighting conditions on both rural and urban environments.

The text in this chapter is adapted and abbreviated from [1–3].

1 INTRODUCTION

General for all computer vision systems to work is that they all require some input video data, as previously mentioned this has a direct impact on the quality and the result of the following stages in the entire pipeline. The primary goal in Video Acquisition is thus to acquire data. As the stage and topic name suggests, this work only consider data in the sense of video, but modern computer vision is heavily relying on other types of sensors than for instance the popular RGB cameras. Making a vehicle drive by itself, for instance, requires a large variety of sensors to operate in an ever dynamic environment, that is, at least until vehicle-to-vehicle communication and vehicle-to-infrastructure communication are fully rolled out. However, even then, and in particular in the time in-between, sensors are expected to play a role as RGB cameras work similar to human vision in the sense that it also relies on some lighting source, e.g., the sun, to be useful. However, given some lighting source RGB cameras will in most cases enable the computer to perceive its environment. The strict requirement of a lighting source challenges the suitability of the RGB camera for especially traffic surveillance and traffic analysis as accidents occur both at daytime and nighttime. This open up for other sensors such as thermal cameras, which are basically a sensor that captures the infrared radiation emitted by all objects relative to its own temperature, which in a more high-level abstraction can be translated to "seeing the temperature" making it useful doing night-time as seen in Figure 2.2 as well as Figure 1.3 [1].



(a) **(b)**

Fig. 2.2: Data collection at 02:00 in the night using two modalities. The (a) RGB camera has many problems in this lighting conditions, whereas the (b) thermal camera only senses the temperature making it more usable in large-scale studies. *Image source: [1].*

The outcome from an RGB camera, or a thermal camera, is a stream of 2D images which constitutes a video when recorded. The two dimensions, i.e., width and height, of the images are often referred to as the resolution of the video input, e.g., 640x480, 1280x1024, 1920x1080. A sensor continuously captures the images with a constant speed often described as frames per second (FPS), which is merely the number of images that the camera captures each second. One of the limitations of the stream of 2D images is, in particular for the self-driving vehicle applications, that it does not provide any exact information about the depth of the scene presented to it. In such cases, other or additional complimentary sensors can be installed. For self-driving vehicles, the depth information is essential to navigate safely in a dynamic scene. Self-driving vehicles have thus installed a large set of sensors besides a regular camera, e.g., stereo cameras, radars, LIDAR. These types of sensors provide the computer with information about the distances to the objects around it, which significantly reduces the risk of the vehicle hitting surrounding obstacles, e.g., pedestrians or cyclist. Relying solely on these types of sensors, would in most cases challenge the computer in respect to assigning a class label to an object. RGB cameras will in most applications complement the other sensors by adding color cues to the object classification rather than only a point cloud.

In the same way, that sensors can compliment each other by providing different information, using multiple sensors can also aid in the sense that the computer does not miss any vital information. For self-driving vehicles, one thing is to get both RGB data and depth data, another just as important thing is to make sure that the vehicle has sufficiently strategically placed sensors such no information is lost, e.g., cyclists in the blind spot or over-taking vehicles. In the same way, automated traffic analysis are heavily relying on all objects of interest, e.g., cars, trucks, pedestrians, and cyclists, are captured and present in the data; otherwise the quality of following automated traffic analysis is reduced correspondingly. The most frequent challenge regarding capturing traffic surveillance data is occlusion as a result of the capturing view-angle. An illustrated of occlusion caused by installing cameras in an existing traffic light pole is seen in Figure 2.3a.

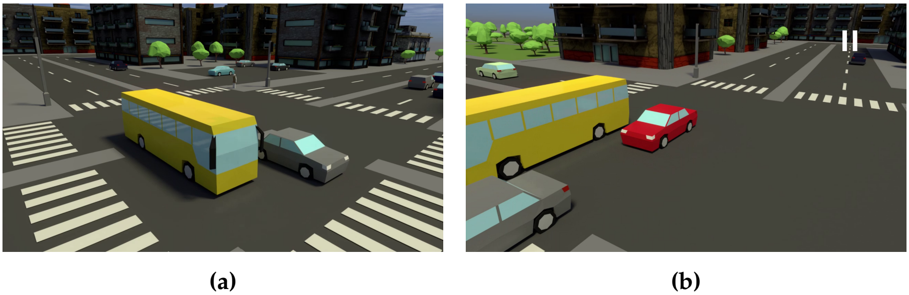

(a) **(b)**

Fig. 2.3: Illustration of how the (a) video stream from a camera mounted in existing infrastructure can have its (b) occlusion challenges due to limited view-angle. *Image source: [2].*

In the example shown in Figure 2.3, the occlusion could be prevented or reduced by installing the camera equipment in a higher altitude. However, this is often not possible due to limited existing infrastructure at the intersections. Other possibilities, therefore, involve deploying a portable pole, if such exists. In most cases, the portable pole can be placed in a more optimal view-angle compared to the existing unmovable infrastructure. Further, the portable pole can be manufactured with an ideal height for capturing data at traffic intersections.

However, if the recording height is in fact the main cause for occlusion, why do we not use drones then? This is often the question raised by traffic researchers, especially after drones have become cheaper and just as easy to operate as playing video games. Using drones are, however, not practical for large-scale studies such as the InDeV project, but say we could in fact make them practical to use. Are the drones then better and the answer to all of our problems as sometimes claimed by traffic researchers, or will other problems occur such as the one illustrated in Figure 2.4.

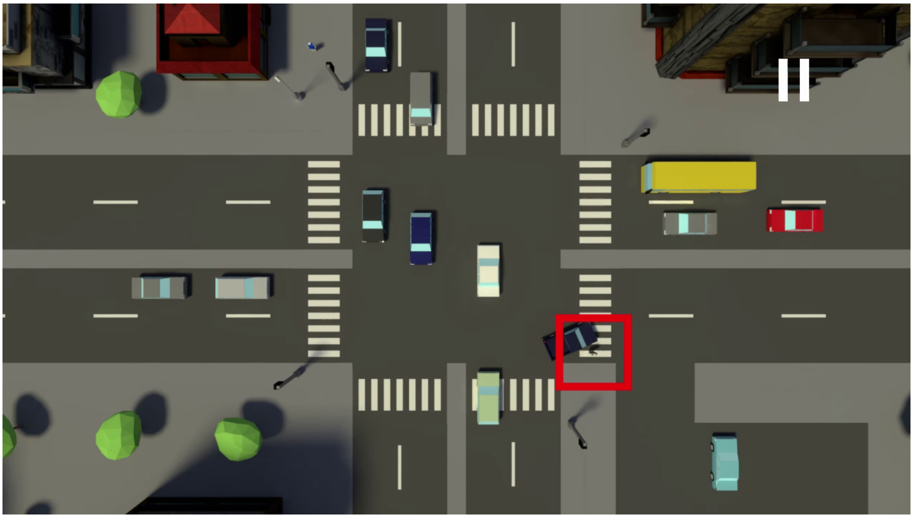


Fig. 2.4: Illustration of how drones provide a very nice overview of the traffic intersection, but due to regulation, the drone is restricted to a minimum height, causing some objects to be small and more difficult to detect in this view-angle, in this case, pedestrians (Marked with red square). *Image source: [2].*

This part of the thesis, will first of all take care of the need of a video acquisition setup in respect to automated traffic analysis within the InDeV project, where 21 traffic intersections across Europe have been recorded continuously for 3-weeks straight. The developed video acquisition setup cannot always rely on being installed in existing infrastructure, why this part also addresses the need of portable poles for acquiring data at both rural and urban traffic intersections. Finally, this part of the thesis will look into the question raised by the traffic researches, and even some computer vision researchers, that by merely adding height to the video acquisition setup, e.g., portable poles or drones, a much better result is reached in respect to automated traffic analysis.

2 STATE OF THE ART

Acquiring video data are, as previously mentioned, a very application dependent task. There do therefore not exists any strict state-of-the-art on how to do it. However, the overall objectives follow some general consideration about the acquisition system as outlined in [3] (chapter A), i.e., multi-view and multi-modal, power supply, vandalism prevention, and recording capacity. The video acquisition setup can thus consist of one or multiple sensors which could be of different modalities. Popular sensors for traffic surveillance are the mentioned RGB cameras and thermal cameras [4–8], but depth sensors such as radar and LIDAR are used in applications where velocity and speed measurements are of interest [9–11].

The sensors can be installed in different existing infrastructure; however, this might not always serve as an ideal view-angle. Using portable poles, or other deployable equipment for that matter, such as the Miovision Scout [12] which is easy deployable as seen in Figure 2.5. The Miovision Scout is very lightweight to transport while being easy and fast to deploy in up to 6.4 meters.

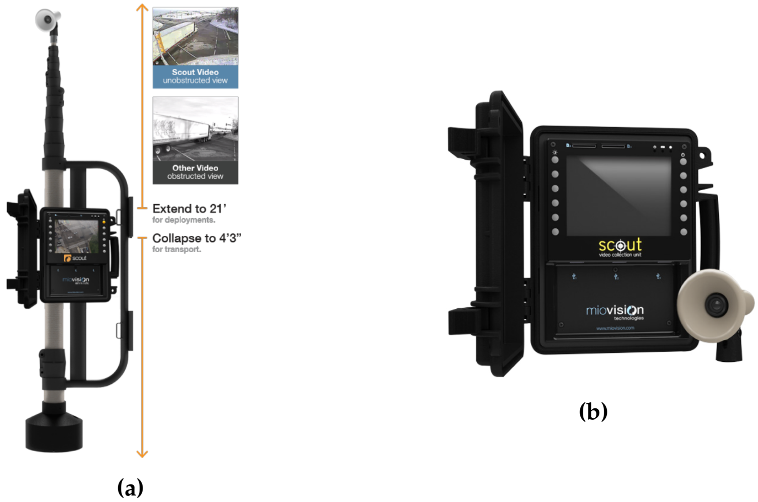
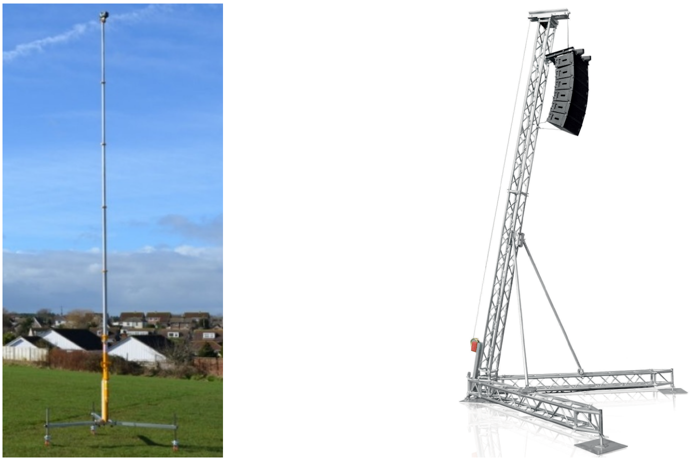


Fig. 2.5: The lightweight and compact portable pole from Miovision. (a) Miovision Scout with the extendable pole (b) Miovision Scout Video Collection Unit and camera. *Image source: [12].*

The Miovision Scout is an ideal portable pole for small studies such as pilot tests, but as the sensor is pre-defined from the manufacturer and thus not configurable, it does not constitute a good platform for large-scale studies. Given that the portable pole should be configurable to a broader set of sensors constituting a more useful video acquisition platform, portable poles consisting of telescopic masts can be used. A lightweight 5-section telescopic Clark QT mast [13] does to some extent keep some of the same pros as the Miovision Scout solution while being configurable for multiple sensors and reaching a capturing height of 10 meters. The 5-section telescopic mast is seen in Figure 2.6a. However, for the mast to remain stable during windy conditions, a guying system is required which may be difficult to install in urban areas.



(a) **(b)**

Fig. 2.6: More configurable portable poles can be based on the (a) lightweight Clark QT Mast on a tripod or (b) heavyweight Litec 7.5-500 Flyintower. *Image source: [13] and [14], respectively.*

Instead of using a lightweight mast with the guying system, a heavyweight mast like the Litec 7.5-500 Flyintower [14] can be used instead. The Flyintower is probably best known as a sound tower seen at concerts lifting the loudspeakers as depicted in Figure 2.6b. However, both the lightweight and especially the heavyweight occupies quite a large area on the ground, which in some urban areas may occupy too much of the pavement and introduce a safety issue as pedestrians need to walk around it. Compared to the Miovison Scout, both the lightweight mast and heavyweight mast requires a large trailer to transport the setup around. Instead of dissembling everything and put it into a trailer, the Swedish based company Trivector has developed the Trivector Mobile mast, which has since been acquired by Lund University. The mast is installed directly in the large trailer and can be extracted to a height of up to 15 meters as seen in Figure 2.7a. This configuration also allow for a higher degree of freedom with respect to installing multiple sensors as seen in Figure 2.7b.

In the same way, the National Aeronautics and Space Research Center of the Federal Republic of Germany (DLR) has developed the Urban Traffic

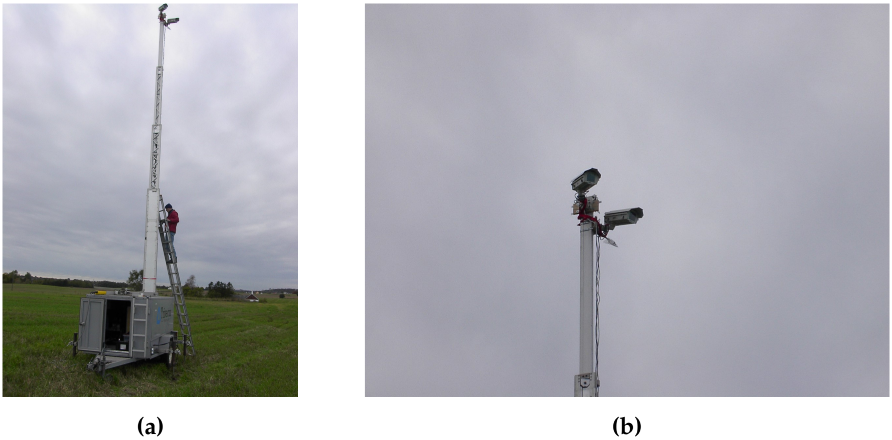
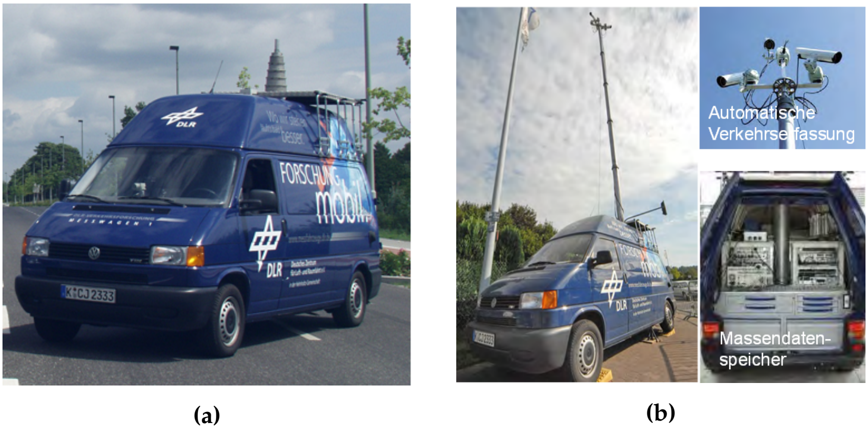

(a) (b)

Fig. 2.7: (a) The Trivector Mobile mast setup in operation. (b) The Trivector Mobile mast with two installed cameras. *Images provided by Aliaksei Laureshyn, Lund University [1].*

Research CAR (UTRaCAR), which is seen in Figure 2.8a in transportation mode and in Figure 2.8b where the left image shows an image of the car in operation [15].

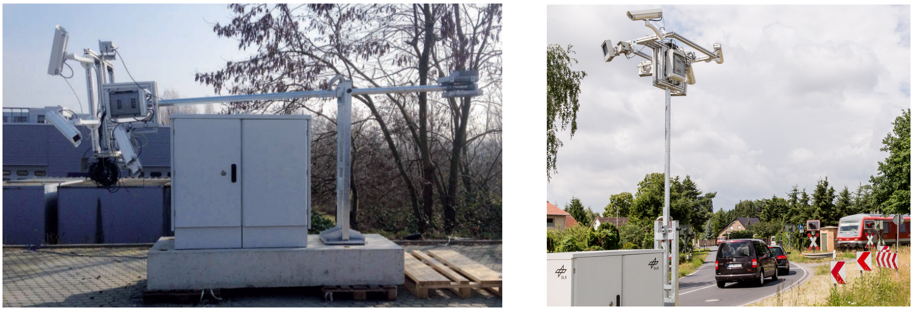


(a) (b)

Fig. 2.8: The National Aeronautics and Space Research Center of the Federal Republic of Germany (DLR) has developed the Urban Traffic Research CAR. In (a) The UTRaCAR is seen with the telescopic mast retracted. (b) The DLR UTRaCar with the telescopic mast extracted. *Image source: [15].*

The deployment of both the Trivector and the UTRaCAR thus faster than the two previous solutions and can easily be equipped with several sensors, but is considered a significantly more expensive solution. Further, both the Trivector mast and the UTRaCAR solution occupies an even large area on the ground, which provides the same challenges as for the lightweight mast and heavyweight mast.

In addition to the UTRaCAR, DLR has also developed the Test field AIM (Application Platform for Intelligent Mobility) to facilitate the acquisition of data. Compared to the UTRaCAR, the AIM platform is built on large concrete platform as seen Figure 2.9a.



(a)

(b)

Fig. 2.9: The National Aeronautics and Space Research Center of the Federal Republic of Germany (DLR) has developed the Test field AIM (Application Platform for Intelligent Mobility) which is seen in (a) transportation mode and (b) operation mode. *Image source: [16].*

In Figure 2.9b the DLR AIM platform is seen in operation mode, where the mast itself reaches an operational height of approximately 4-5 meters [16]. This is considered a rather heavy platform; thus a truck and crane are needed in order to transport and then actually deploy the platform at a given point of interest.

In recent years, drones have become more accessible and the flight time of them has allowed staying in the air for more than 30 minutes. This opens up the possibility of using drones as a platform for video acquisition in respect to automated traffic analysis and traffic surveillance, which is already heavily used [17–21]. There is a public believe, that the drones are the solution to most of the problems and is a future-proof technology [17]. Most research does indeed show promising results for vehicle detection, e.g., cars, trucks, and buses, as a result of the low degree of occlusion and excellent overview of the traffic intersection [22, 23]. Detection of pedestrians, on the other hand, is a more challenging problem. In [24] a drone dataset is captured in respectively 2, 3, and 4 meters capturing height, which by applying four different algorithms based on Haar-LBP, Haar, LBP, and HOG, respectively, all show a decrease of detection rate as altitude increases. The best performing algorithm, Adaboost with HOG features, reaches a detection rate of 67.23 %, 63.72 %, and 60.44 % on respectively the 2, 3, and 4 meters capturing height data.

In [25] the pedestrian detection from a drone is carried out by the usage of a cascade of boosted classifiers which are based on the well-known Haar-like features [26, 27]. To verify the system, they collect a small multi-modal RGB and thermal dataset with 11 humans in it. Though the system does detect the 11 humans, it also generates false positives especially when the capturing altitude increases. In [28], the combination of RGB and thermal are used again with cascades boosted Haar classifiers. They report a detection rate for pedestrians and cars on 70 % and 80 %, respectively. A detection in this study translates to detecting the unique object at least once in the data stream rather than detecting it in every image. Relying only on thermal data for pedestrian detection is done in [29], which show decent results in a small study using blob extraction on thermal video.

Though it is easy to claim, that pedestrian detection using thermal images from UAV is straightforward to solve. What happens when the data are captured at a traffic intersection with other objects looking similar from the bird-view angle, e.g., cyclists and motorcycles. Especially when considering the decreasing detection rate when altitude increases as described in [24]. In [30] the VIRAT dataset is assembled, which contain data captured from both stationary ground cameras as well as moving aerial vehicles, i.e., a manned aircraft, at different locations such as parking lots and around buildings. The same paper describe how the aerial dataset, though it is from a moving aircraft, serves as a more significant challenge due to the changing viewpoints and occlusion issues. However, they do not explicitly capture data at a traffic intersection at the same point of time allowing a fair comparison between the two view-angles. There does not exist quantitative nor qualitative research in respect to traffic analysis at traffic intersection that focuses on whether drone view-angle being "better" than the existing infrastructure view-angle - or what they are "better" at and when.

3 CONTRIBUTIONS

During this Ph.D., three works have been carried out concerning capturing video data for automated traffic analysis at traffic intersections. Two of the investigated works are both heavily linked to the InDeV project where a proper video acquisition setup was required in order to capture continuous data throughout 3-weeks and one year when attached to an infrastructure power supply. We have thus developed a recording platform for video acquisition at traffic intersection. The recording platform consists of two systems allowing two view-angles. Each system consists of same basic recording equipment which are a Network Attached Storage server, network switch, wireless access point, and batteries. All of the recording equipment is installed in an aluminum box which can be chained to existing infrastructure such the risk of vandalism and theft are minimized. The two systems can be the use of the wireless access points, which are installed outside of the aluminum box, be connected using a wireless connection, thus allowing time synchronization using Network Time Protocol. The main difference between the two setups is that one of the setups makes use of both an RGB camera and a thermal camera and the other only has an RGB camera. The development of the InDeV recording setup has led to a technical report used internally within the InDeV project [3] (chapter A).

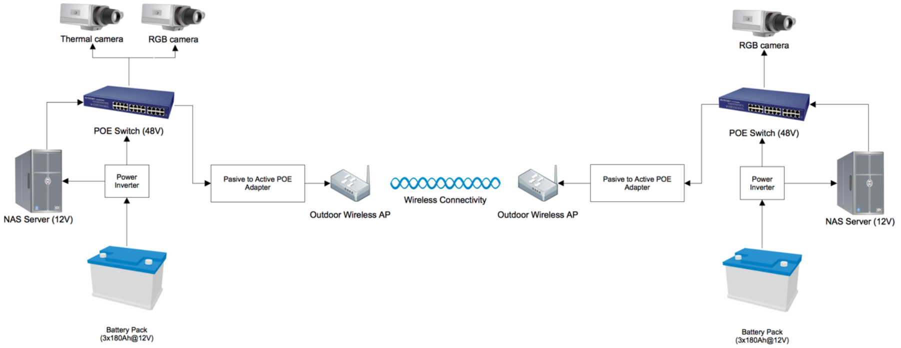


Fig. 2.10: Overview of the hardware and its connectivity within the InDeV recording setup
Image source: [3].

The developed InDeV recording setup has been used to capture several months of data in Europe. An overview of the hardware and how it is connected in the developed InDeV recording setup is seen in Figure 2.10.

One of the shortcomings of most video acquisition setups, including the InDeV recording setup, is that it is relying on existing infrastructure. In [1] (chapter B), we survey four general types of portable poles which show a lack of lightweight, robust, and yet mobile portable pole. We therefore propose and develop a new portable pole satisfying these requirements in [1]. The developed portable pole has been put to use several times at rural locations where an example is seen in Figure 2.11. In rural locations, there is often many limitations with respect to infrastructure to install video acquisition equipment in.

Additionally, the developed portable pole has seen its advantage in urban environments due to its low ground area occupation. The survey of the portable pole was initially drafted in 2016 as an internal technical report for the InDeV project. Through 2016 to 2018, the development and usage of the new portable pole were created, which was ultimately combined with the survey and published in *Journal of Transportation Technologies* in 2018 [1]

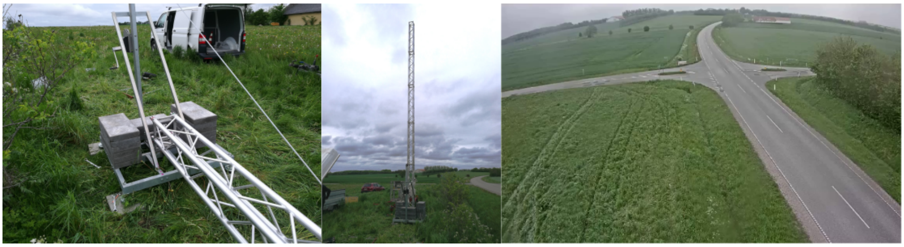


Fig. 2.11: The TRG-pole is deployed at a traffic intersection with limited existing infrastructure. *Image source: [1].*

(chapter B).

Finally, to address the pros and cons of using a drone as a video acquisition platform in respect to traffic analysis at intersections. We have collected a synchronized multi-view dataset from respectively existing infrastructure and drone which eventually will become freely available as described in [2]. The dataset consists of 3,100 frames from each view-angle. Each frame is annotated with both axis-aligned bounding box as well pixel-level annotations. An initial baseline experiment has been carried out by evaluating the two view-angles using a pre-trained Mask R-CNN object detector for axis-aligned bounding boxes. The preliminary results show that the mean average precision (mAP) are 22.62 % and 27.75 % for the existing infrastructure view-angle and the drone view-angle, respectively. This indicates that the drone view-angle is in fact superior to the existing infrastructure, which is also the case if only considering cars, bus, and lorries. If we, however, look at cyclists, none are detected in the drone view-angle. This work, as presented in [2] (chapter C), is still ongoing research and is expected to grow into a journal in 2019.

The main contributions of this Ph.D. thesis within the field of Video Acquisition are thus:

- Developed a multi-view and multi-modal recording platform for video acquisition at traffic intersection. The recording platform has captured several months of data both in Denmark and Belgium.
- We have surveyed and provides an overview of four general types of portable poles with respect to capturing data at traffic intersections in rural and urban environments.
- The portable pole overview pointed out the lack of a lightweight, robust yet mobile portable pole. We have thus designed and developed a new portable pole satisfying these requirements.

- We have collected and annotated a synchronized multi-view traffic intersection dataset allowing performance comparison between video acquired with a drone and video acquired by installing camera equipment in existing infrastructure, respectively.
- Preliminary result based on a pre-trained Mask R-CNN object detector shows that the drone view-angle is superior to the existing infrastructure view-angle when looking for larger objects, e.g., cars, trucks, and lorries, but when looking at pedestrians and cyclists, the roles are flipped as none of these are detected in the drone view-angle.

REFERENCES

[1] M. B. Jensen, C. Holmberg Bahnsen, H. S. Lahrmann, T. Madsen, and T. Moeslund, “Collecting traffic video data using portable poles: Survey, proposal, and analysis,” *Journal of Transportation Technologies*, vol. 08, pp. 376–400, 01 2018.

[2] M. B. Jensen and T. Moeslund, “Multi-view traffic intersection dataset: Performance analysis and comparison,” *Ongoing*, 2018.

[3] M. B. Jensen and T. Moeslund, “Indev recording setup,” *InDeV Technical Report*, 2018.

[4] A. Laureshyn, “Application of automated video analysis to road user behaviour,” 2010.

[5] T. K. Madsen, C. Bahnsen, H. Lahrmann, and T. B. Moeslund, “Automatic detection of conflicts at signalized intersections,” in *Workshop on the Comparison of Surrogate Measures of Safety Extracted from Video Data, Transportation Research Board 93rd Annual Meeting*, 2014.

[6] C. Bahnsen and T. B. Moeslund, “Detecting road user actions in traffic intersections using rgb and thermal video,” in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[7] C. Bahnsen and T. B. Moeslund, “Detecting road users at intersections through changing weather using rgb-thermal video,” in *International Symposium on Visual Computing*. Springer, 2015, pp. 741–751.

[8] J. Xu, C. Fookes, and S. Sridharan, “Automatic event detection for signal-based surveillance,” *CoRR*, vol. abs/1612.01611, 2016. [Online]. Available: http://arxiv.org/abs/1612.01611

[9] J. M. Munoz-Ferreras, F. Perez-Martinez, J. Calvo-Gallego, A. Asensio-Lopez, B. P. Dorta-Naranjo, and A. B. del Campo, "Traffic surveillance system based on a high-resolution radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1624–1633, June 2008.

[10] A. Börcs and C. Benedek, "Extraction of vehicle groups in airborne lidar point clouds with two-level point processes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1475–1489, 2015.

[11] M. S. Shirazi and B. T. Morris, "Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2017.

[12] Miovision. Scout video collection unit. [Online]. Available: https://miovision.com/scout/

[13] C. Masts. Clark masts qt series. [Online]. Available: http://www.clarkmasts.com/products/telescopic-masts/qt-series/

[14] Litec Strutture & Soluzioni. Flyintower 7.5-500 catalogue. [Online]. Available: http://www.litectruss.com/Litec/media/litec/Downloads/FLYINTOWER_7-5-500_catalogue.pdf?ext=.pdf

[15] M. Junghans, "Situations- und gefahrenerkennung in verkehrsszenen," in *Kolloquium Verkehrsmanagement und Verkehrstelematik*, Dresden, Germany, May. 8 2013. [Online]. Available: http://www.vimos.org/cms/data/uploads/termine/junghans_sgv.pdf

[16] D. I. of Transportation Systems, "Aim mobile traffic acquisition: Instrument toolbox for detection and assessment of traffic behavior," *Journal of large-scale research facilities*, no. 2, A74, 2016.

[17] M. A. Khan, W. Ectors, T. Bellemans, D. Janssens, and G. Wets, "Uav-based traffic analysis: A universal guiding framework based on literature survey," *Transportation Research Procedia*, vol. 22, pp. 541 – 550, 2017, 19th EURO Working Group on Transportation Meeting, EWGT2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352146517301783

[18] J. Apeltauer, A. Babinec, D. Herman, and T. Apeltauer, "Automatic vehicle trajectory extraction for traffic analysis from aerial video data," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 3, p. 9, 2015.

[19] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *Unmanned*

Aircraft Systems (ICUAS), 2013 International Conference on. IEEE, 2013, pp. 221–234.

[20] A. Puri, "A survey of unmanned aerial vehicles (uav) for traffic surveillance," *Department of computer science and engineering, University of South Florida*, pp. 1–29, 2005.

[21] G. Salvo, L. Caruso, and A. Scordo, "Urban traffic analysis through an uav," *PROCEDIA: SOCIAL & BEHAVIORAL SCIENCES*, vol. 111, pp. 1083–1091, 2014.

[22] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car detection from low-altitude uav imagery with the faster r-cnn," *Journal of Advanced Transportation*, vol. 2017, 2017.

[23] A. Pérez, P. Chamoso, V. Parra, and A. J. Sánchez, "Ground vehicle detection through aerial images taken by a uav," in *Information Fusion (FUSION), 2014 17th International Conference on*. IEEE, 2014, pp. 1–6.

[24] W. G. Aguilar, M. A. Luna, J. F. Moya, V. Abad, H. Parra, and H. Ruiz, "Pedestrian detection for uavs using cascade classifiers with meanshift," in *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, Jan 2017, pp. 509–514.

[25] P. Rudol and P. Doherty, "Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery," in *2008 IEEE Aerospace Conference*, March 2008, pp. 1–8.

[26] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[27] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[28] A. Gaszczak, T. P. Breckon, and J. Han, "Real-time people and vehicle detection from uav imagery," in *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, vol. 7878. International Society for Optics and Photonics, 2011, p. 78780B.

[29] Y. Ma, X. Wu, G. Yu, Y. xu, and Y. Wang, "Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery," *Sensors (Basel, Switzerland)*, vol. 16, 03 2016.

[30] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy,

M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, “A large-scale benchmark dataset for event recognition in surveillance video,” in *CVPR 2011*, June 2011, pp. 3153–3160.

Chapter 3

Object Detection

Object detection is the second main topic after Video Acquisition in this Ph.D. thesis as seen from Figure 3.1. Object detection in this thesis is mainly concerning traffic light detection (TLD) but do also include object detection for automated traffic analysis. This chapter will thus mainly provide an overview of TLD and the corresponding state-of-the-art. Though this chapter is mainly addressing the challenges of TLD, i.e., localizing the traffic lights, this chapter will also touch challenges related traffic light classification, which combined with TLD, constitutes traffic light recognition (TLR).

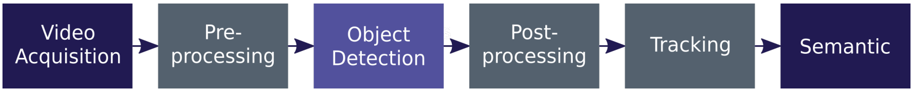


Fig. 3.1: Object detection is the second main topic covered in this Ph.D. thesis.

TLR is a research topic that is important for autonomous vehicles as well as for driver assistance systems (DAS). The core methods used for both TLD, automated traffic analysis, and most other object detection problems, are the same. This chapter is thus not going to distinguish between the object detection methods and their applications when describing the state-of-the-art.

Large parts of this chapter are adapted, edited, and abbreviated from [1] and [2].

1 INTRODUCTION

Object detection is the ability to localize a given object within a data input such as an image or video. Commonly the localizing has been done by marking the object of interest with the axis-aligned bounding boxes. This is often considered the critical stage in a computer vision as the ability to detect an object is the primary reason to apply computer vision for most people. The applications for object detection can be many; popular applications include as face detection, pedestrian detection, vehicle detection, traffic sign detection, etc. However, it could also be some more specific objects, e.g., computer vision system can be used in food quality control spawning the need of detecting diseases on the organs. In such applications, detection on pixel-level such instance segmentation may be more suitable. The requirements of object detection are very application dependent which also define some application depended challenges as we shall see.

Automated TLR is as previously mentioned the ability of making a computer vision system able to localize and determine the state of the traffic lights in intersections. In this specific chapter, we consider TLR in respect to driving assistance systems which at some point is expected to aid in the progress towards autonomous vehicles. However, TLR can serve a broad set of applications:

1. DAS: Aid the driver by either providing information about the state of the approaching traffic intersection or by partly control of the vehicle.

2. Autonomous vehicles: Making vehicles drive completely autonomous are a multi-disciplinary task, e.g., communication between both vehicle-to-vehicle as well as vehicle-to-infrastructure, which may make TLR obsolete one day. However, up until then, making a vehicle operate autonomously in a dynamic environment requires it to understand its surrounding using sensors. Especially at traffic intersections where accidents are more prone. By the use of TLR, this can be achieved.

3. Automated traffic analysis: In a later stage of the computer vision pipeline is Semantics, as seen in Figure 3.1. Detecting vehicles running a red light is of high interest for traffic researchers as this often have critical consequences. In addition to having a decent vehicle detector, TLR is required in order to make this information available at a larger stage in the computer vision pipeline.

A general high-level breakdown of a computer vision based TLR system. Before the rise of machine learning and especially deep learning, research in TLR, and in most recognition problems, e.g., traffic signs recognition [3], has generally been split into two steps, namely detection and classification, as seen in Figure 3.2.

In the detection step, the purpose is to locate the traffic light in the input image from the Video Acquisition stage. Classification is the following step, where the localized traffic light is assigned to a class, i.e., a traffic light state. The two steps can be carried out individually or in a combined system. The classification step however depends on the detector localizing the correct objects first. As we shall discuss in the state-of-the-art section, these two steps

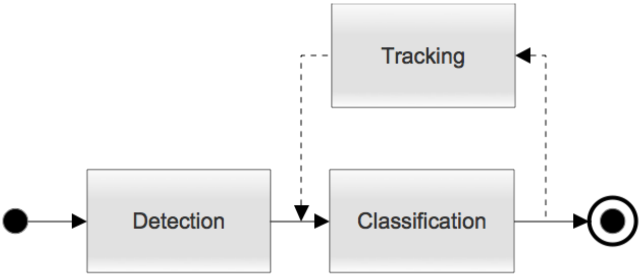


Fig. 3.2: A general high-level breakdown of a computer vision based TLR system. *Image source: [1]*

are a very high-level breakdown and are in some full and newer systems described as one. In this thesis, the general computer vision pipeline, seen in Figure 3.1, also, for simplicity sake, define this as a somehow combined step. For some applications, it may make sense to adjust this to a more detailed pipeline. In this chapter the focus will be on the detection step.

In most case, the traffic lights consist of primarily three states, i.e., green, yellow, and red, which corresponds to go, stop if possible, and stop. For most simple traffic intersection, the combination of these three traffic light states suffices. However, when developing more complex traffic intersections, which are often the case in urban environments, several variations of the traffic light control devices have been introduced. In Figure 3.3 a subset of some of the frequently used traffic control devices.

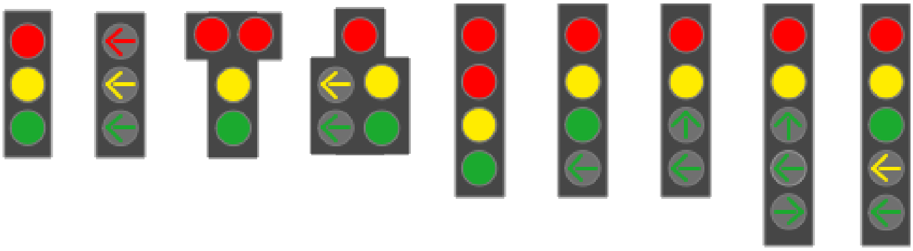
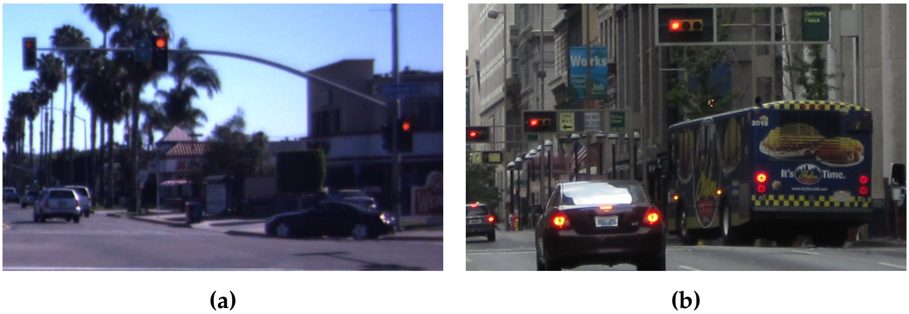


Fig. 3.3: Examples showing a subset of the conventional vertical traffic lights used in complex traffic intersection in California, USA. *Image source: [4].*

Even though there are some well-defined standards and conventions of how traffic lights look, e.g., California, USA as seen in Figure 3.3, this does not necessarily apply to other countries as the variations can vary quite a lot from continent to continent, country to country, and even from state to state as evident in Figure 3.4. A fully automated TLR system must be quite robust to the many variations.

Besides the orientation of the traffic light, there exists a lot of challenges



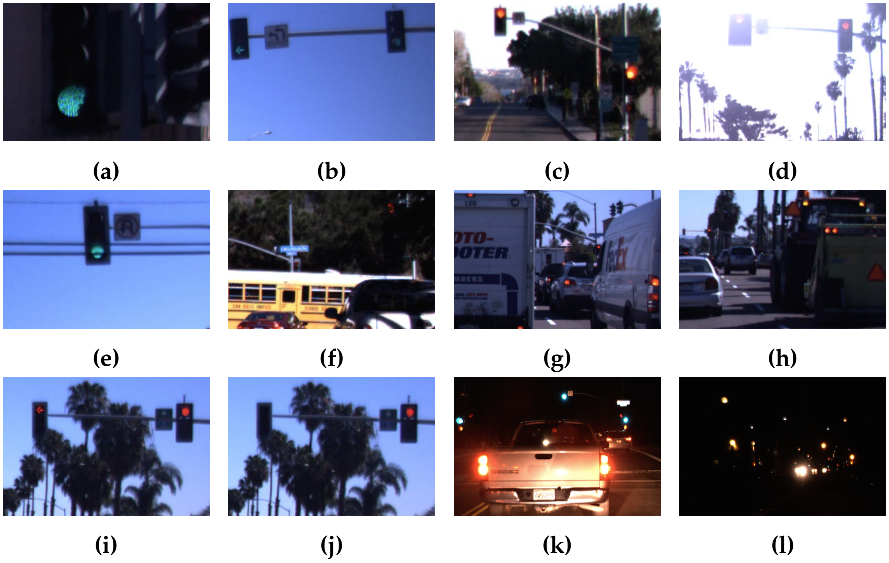
(a) **(b)**

Fig. 3.4: The orientation of the traffic lights can vary a lot even within the same country. (a) San Diego, California. (b) Cincinnati, Ohio. *Image source: [1, 5, 6]*

and issues when developing TLR There are several issues for detecting traffic lights in urban environment as well as rural, most common issues are listed in [1, 5, 6]:

- *Color tone shifting and halo disturbances [7], e.g., because of atmospheric conditions of influences from other light sources. Figures 3.5c, 3.5d, 3.5k and 3.5l.*
- *Occlusion and partial occlusion because of other objects or oblique viewing angles [7]. Especially a problem with supported TLs [8–10]. Figures 3.5e to 3.5g.*
- *Incomplete shapes because of malfunctioning [7] or dirty lights. Figure 3.5a.*
- *False positives from, brake lights, reflections, billboards [11, 12], and pedestrian crossings lights. Figure 3.5h.*
- *Changes in lighting due to adverse weather conditions and the positioning of the sun and other light sources.*
- *Synchronization issues between the camera's shutter speed and TL LED's duty cycle. Figures 3.5i and 3.5j.*

Before TLR can be fully adapted in autonomous vehicles, all of these issues and challenges must be addressed and solved. Further, for a TLR system even to be useful, it must be able to assess the traffic light at a far distance in order to react upon the outcome in a timely manor. This is however not straightforward as complex traffic intersection may be filled with several traffic lights with different purposes, and most of them are not relevant for the autonomous vehicle or the driver if we consider TLR as a DAS. Making TLR useful and well working from a user's point of view requires the TLR to work together with several other applications, e.g., lane estimation, traffic sign recognition, and vehicle detection. Further, the vehicle needs to have an



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 3.5: Common issues and challenges in the quest of developing TLR systems. *Image source: [1, 5, 6].*

idea of how to select the relevant information as it will be exposed to a lot of it. For example, which traffic light is the most relevant one in Figure 3.6? There are traffic lights in the scene from three different traffic intersections which are located within a short distance on the same main road. Though it might be in a later stage than Object Detection, a complete TLR system must be able to determine this.

Besides looking at traffic lights, this chapter also briefly touch upon detecting vehicles and other objects in respect to ultimately doing automated traffic analysis. The problem is a little different as the view-angle of the used data is changed from the moving vehicle to a static capturing angle as illustrated in Figure 3.7.

Traffic analysis is a Semantic stage, but in order to carry out any Semantic analysis, good object detection is needed. The most common objects of interest in relation to traffic analysis are different vehicles classes such as cars, trucks, and lorries [14]. However, one of the reasons of performing a traffic analysis is often to prevent fatal accidents at especially urban areas such as traffic intersections. According to [15], 1.2 million people lost their lives on the roads globally in 2017. To put that into perspective, that corresponds to the ninth biggest cause of death globally. In the same study, it is reported that 49 % of those victims are vulnerable road users such as pedestrians, cyclists, moped riders, and motorcyclists. In order to ultimately prevent traffic accidents, several different objects are thus required in order to do useful traffic

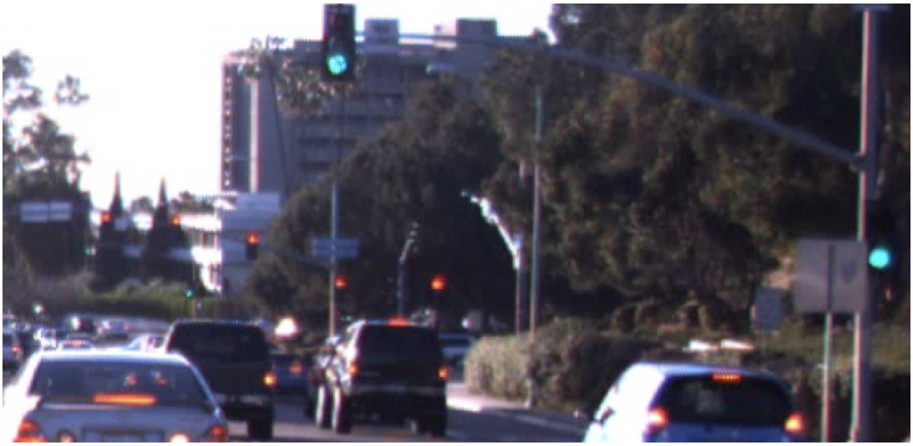


Fig. 3.6: Several traffic intersections can be located within a short distance as evident in this example from San Diego, USA. Three traffic intersections are located within a few hundred meters on the same main road. *Image source: [1, 5, 6]*

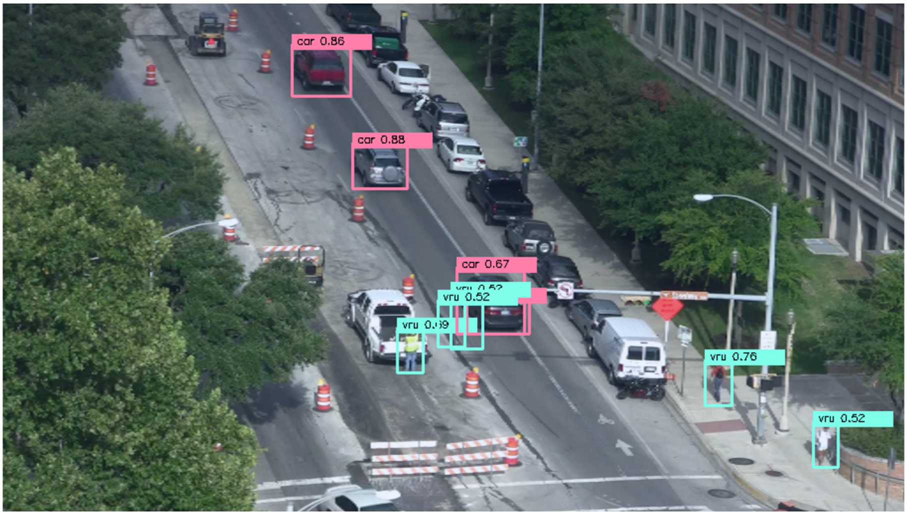


Fig. 3.7: Detecting objects, e.g., cars and vulnerable road users, are a vital part of doing automated traffic analysis. *Image source: [13]*

analysis.

2 STATE OF THE ART

Detecting objects, whether that is traffic lights, cars, pedestrians or cyclists, do to some extent utilize the same methods. Most of the modern object detections methods are in fact benchmarked on datasets that consists on a large variety of classes, e.g., PASCAL VOC [16], MS COCO [17], VIRAT dataset [18], and ImageNet ILSVRC [19]. These datasets have played an essential role in the development and advancement of object detection. Though they vary in size and class depth, they do constitute an open-source and common benchmark platform, which have been vital for a fair comparison between the many object detection methods.

Before the introduction of these larger datasets, most research in TLD has been evaluated on the researchers' own private and often small datasets [1, 5]. In 2010 the public TLR benchmark dataset was made available by the *Robotics Centre of Mines ParisTech* in France [20]. The dataset is captured in an urban environment and contain 11,179 frames hand-labeled with 9,168 axis-aligned bounding box annotations. Unfortunately, the usage of the dataset has been limited.

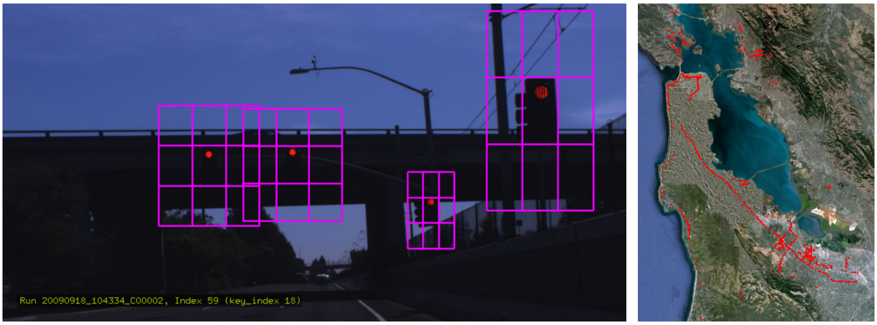
The detection of traffic light can be split into two categories, namely heuristically model-based methods and learning-based methods. Up until around 2015, the heuristically model-based methods have been dominated the field. Common methods rely on some heuristically determined model based on threshold from, e.g., specfic shapes, color distributions, and intensities. The most simple and surprisingly widely used one rely on defining some lower and upper color intensity threshold for each of the three traffic light color states as suggested in [7, 11, 12, 21–29]. This works well given some very non-dynamic data, as the thresholds defined are often optimized and overfitted towards one specific dataset. To make this type of models a little more robust to changing scene, [8, 30] investigates several detectors whereas a Gaussian pixel classifier based on thousands of manually annotated traffic lights is the detector that achieves the best result. The usage of Gaussian Mixture Models (GMM) has also seen its use in for automated traffic analysis where an adaptive background is estimated and vehicles and pedestrians moving through the scene do not match any of the calculated background models [14].

The main drawback of these detectors solely based on color is that the color intensity is subject to variation as a result of the varying lighting conditions in the scene as illustrated in some the samples in Figure 3.5.

Besides the color cue, a traffic light is often characterized by its shape. Popular approaches involve looking for Hough transform on edge map [26, 29, 31] as well as the faster radial symmetry to find circles [24, 32].

The learning-based methods are a data-driven approach to construct the models. Rather than heuristically deriving the correct characteristics for a given object, the learning-based methods derive these characteristics automatically by being exposed to large quantities of data. In [8, 30] a cascading classifier created on Haar-like features [33] were introduced for TLD. The cascaiding Haar classifier has however seen significant use in face detection [34], pedestrian detection [35], vehicle detection [36], and probably most detection domain present. In [37, 38], Haar features are fused with GMM constituting a pedestrian detector used to carry out behavioral analysis at traffic intersections. In 2005, the Histograms of oriented gradients (HOG) features combined with support vector machines (SVM) were introduced for pedestrian detection [39]. The combination of HOG features and SVM have since been applied and used in most detection applications including TLD [40] and vehicle detection [41, 42] as well as vehicle/pedestrian classification [38]. [40] actually reaches precision and recall of 92.3 and 99.0 % on their own private dataset of 9,301 frames. In 2015, we proposed utilizing the Aggregated Channel Features (ACF) together with learning [43, 44] (Chapter G and E) and [45] trees on the new public available LISA Traffic Light dataset published with [1] (Chapter D).

Most of the industrials actors researching have been utilize many of the aforementioned learning-based methods including neural networks. However, they use a lot more data to make their decisions, which are critical for self-driving cars, as false positives can have fatal consequences. To reduce the false positives, several industrial partners, e.g., Daimler and Google, make use of the GPS coordinates of the vehicle which is combined annotated prior maps of the scene [8, 11, 30, 40, 46, 47]. The usage of prior maps greatly aids the detection of traffic lights, as information of where in the scene given a GPS coordinate the traffic lights are located makes the search area of traffic light smaller as seen in Figure 3.8. Obviously this requires much information about the scene, but it greatly aids the task of rejecting the false positives as well as accepting true positives.



(a) **(b)**

Fig. 3.8: Prior maps of the scene can help defining the search area as seen in (a). It requires that the traffic lights are annotated in the scene and the GPS coordinate is logged as Google has done in San Jose are in San Fransisco, which is seen in (b). *Image source: [11]*

As previously mentioned, TLD was heavily dominated by heuristically model-based methods before 2015, but have since changed into the data-driven approach based on machine learning. This has been a very general paradigm change for the entire field of object detection. The field of object detection has changed even further in the last few years as a result of the hardware improvements made on particular GPUs. This has provided researchers with even more computational power which allows for a sub-field of machine learning called deep learning. A simplified comparison of how deep learning and machine learning differs is seen in Figure 3.9. The work mentioned above in respect to learning-based methods all includes deriving some features based on some method. This is the main difference between deep learning and traditional machine learning, as deep learning, given large quantities of annotated data, can select the relevant features which characterize the object of interest.

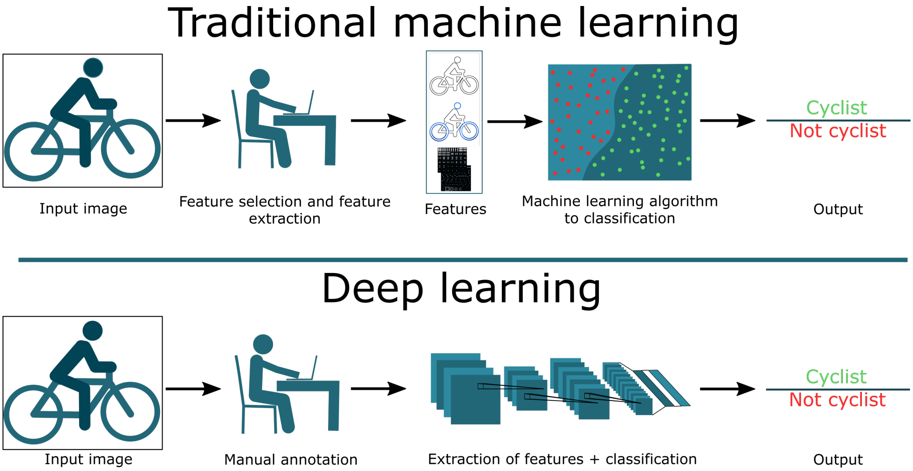

Traditional machine learning
Cyclist
Not cyclist
Input image
Feature selection and feature extraction
Features
Machine learning algorithm to classification
Output
Deep learning
Cyclist
Not cyclist
Input image
Manual annotation
Extraction of features + classification
Output

Fig. 3.9: Simplified comparison of how computer vision for classification using traditional machine learning approach has changed with the introduction of deep learning. *Image source: [2]*

Deep learning is utilizing convolution neural networks (CNNs) which has quickly challenged and outperformed the aforementioned traditional approaches [48, 49]. In 2012 a CNN was applied by Alex Krizhevsky et al. [50] to one of the largest object recognition benchmarks called ImageNet. The dataset consists of 1.2 million training images, 50,000 validation images, and 150,000 testing images. The CNN applied nearly halved the top-5 error rate compared to the traditional methods [49].

Deep learning has since spread rapidly to all of the object detection areas outperforming all of the traditional methods and ultimately defining a new era of object detection. Thus, all of the best-performing methods in other benchmarks such as the MS COCO [17] are learning-based methods. In [51] R-CNN is introduced, which can be characterized by generating approximately 2,000 potential boxes that may or may not contain objects, so-called region proposals, by the use of selective search [52]. From each of these region proposals, the CNN similar to the one introduced with [50] are used to extract the features. Finally, a trained SVM is used to determine whether there is a high probability of each region proposal containing a given class. As one can imagine this process is quite slow as all the approximately 2,000 region proposals are being fed individually to the CNN which then carry out feature extraction, [53] reports the entire process taking up to 47 seconds per image using a GPU. In [53] same authors introduces Fast R-CNN, where the most significant change is the speed up introduced. What makes the Fast R-CNN fast is that the CNN does not take the region proposals as the sole input, it takes the entire input image as input rather than only the approximately 2,000 region proposals. The CNN then calculates a feature map of the entire corresponding input preventing the many overlapping regional proposal computations done in R-CNN. Further, the SVM is replaced with fully connected layers for classification which make Fast R-CNN nearly real-time able. The main bottleneck of Fast R-CNN is the generation of the region proposal from selective search [52]. In [54] the region proposals are substituted with Region Proposal Network (RPN), which calculates the region proposals using a CNN. RPN is combined with the detection network from Fast R-CNN constituting an end-to-end network. Following the naming from the previous iterations of R-CNN, the method introduced in [54] is called Faster R-CNN. Common for all of these is still that they work as a multi-step detector based on region proposals. The Faster R-CNN has been reported to operate with an FPS of 7 on the VOC2017 test set which is not applicable for real-time applications. Rather than having two networks, the first is responsible for the region proposals and second is responsible for the classification, combining these two into one single network. This is done in so-called single-shot detectors such as SSD [55] and YOLO [56–58], where the network proposing regions are replaced with pre-defined boxes which are used to detect the object of interest.

The aforementioned deep learning object detection methods have seen a vast and large use in almost every object detection area such as vehicle detection [59–63] and pedestrian detection [56, 64, 65]. In 2017, [66] we applied the YOLO detector to the LISA Traffic Light dataset, which improved the overall area-under-curve from a calculated precision-recall curve with 50.32 % compared to the previous TLD based on the ACF [45]. In 2017, we also apply the SSD as an object detector in an automated traffic analysis application by adding an additional input layer to the SSD method which we call Bonus Input Layer Single-Shot Multibox Detector (BILSSD). By adding a bonus layer containing temporal information of the scene we improve the accuracy with up to 66 % in our experiment on the VIRAT dataset [18].

3 CONTRIBUTIONS

The work carried out in this part of the Ph.D. Thesis has mainly been in respect to pushing the state of the art in detection of traffic lights. The first major study published was *Vision for looking at traffic lights: Issues, survey, and perspectives* [1] (chapter D). This paper contains the first thorough and comprehensive overview of the current state of the art as well as evaluation within the field of TLR. Three main conclusions were made from this work:

1. Most of the published research within TLR have traditionally been done on small and private datasets. One public available dataset exists but have seen a very limited use.
2. The evaluation of TLR has not been standardized and has thus not been carried out using directly comparable metrics.
3. TLR has not followed the general tendency of switching to more learning-based methods as it clear from the survey that it have been very dominated by model-based methods, especially for TLD.

To aid the overall advancement of working with traffic lights, we thus assembled, annotated, and published the LISA Traffic Light Dataset[1], which contain stereo-video data captured during both daytime and nighttime in San Diego, USA. The dataset consists of 43,007 frames which provides 119,231 annotations split between 304 unique traffic lights making it the largest of its kind. The dataset were initially published in an abbreviated version of the entire traffic light survey at the *International Conference on Advanced Video and Signal-based Surveillance* in 2015 [5]. Further, we define a standardized way of evaluating traffic light detectors in general, but in particular for the LISA Traffic Light Dataset.

In 2015, we began the quest of utilizing learning-based methods to solve the LISA Traffic Light Dataset. The learning-based methods had seen great advancements and use in other similar object detection areas such as vehicle detection, pedestrian detection, and traffic sign detection. We made a learning-based traffic light detector based on ACF, which together with it predecessor ICF had seen great use in pedestrian detection and traffic sign detection. The initial work during was carried out on the nighttime challenge in the LISA Traffic Light Dataset by implementing three different heuristically determined model-based method and comparing that a trained ACF detector. This work was presented at the *Advances in Visual Computing: 11th International Symposium on Visual Computing* in 2015 [43] (chapter G). The investigations and comparison from this publication showed the learning-based methods was a great competitor to the heuristically determined model-based

[1] Available: `https://www.kaggle.com/mbornoe/lisa-traffic-light-dataset/home`

methods and did - not surprisingly - perform better. This brought TLD up to date and on par with research carried out in similar fields such as traffic sign detection and pedestrian detections. To explore the optimal parameters and improve the performance for the nighttime challenge, we carried out a comprehensive parameter sweep adjusting three basic parameters in the ACF which again - not surprisingly - improves the nighttime performance. The outcome is of course a very overfitted TLD that improves the nighttime entry on the dataset with an area-under-curve from a calculated precision-recall curve to 66.63 %. The entire exploration of the adjusting the three basic parameters in ACF was presented at the *Advances in Visual Computing: 12th International Symposium on Visual Computing* in 2016 [44] (chapter E). In 2017, we applied and evaluated the YOLO detector on the daytime challenge in the LISA Traffic Light Dataset. The result of this improved the area-under-curve from a calculated precision-recall curve to 90.49 %, which is an improvement of 50.32 % compared to the previous ACF detector [45]. This work was published at the *Conference on Computer Vision and Pattern Recognition Workshop: Traffic Surveillance Workshop and Challenge* in 2017 [66] (chapter F).

Finally, we investigated adding an additional bonus layer containing compact temporal information to SSD for object detection in respect to traffic analysis in 2017. This work showed that the addition of temporal information improved the accuracy with up to 66 % in experiments on the VIRAT dataset [18]. We presented this work at the *International Conference on Computer Vision Workshop: Computer Vision for Road Scene Understanding and Autonomous Driving* in 2017 [13] (chapter H).

The main contributions of this Ph.D. thesis within the field of Object Detection are thus:

- The public available LISA Traffic Light Dataset which consists of 43,007 frames and 119,231 annotations split between 304 unique traffic lights captured during both daytime and nighttime. The LISA Traffic Light Dataset is the largest of its kind

- In addition to the dataset, we have proposed a standardized way of evaluating TLD on the LISA Traffic Light Dataset easing comparison between methods.

- We have provided a comprehensive overview of the current issues, challenges, used methods, and perspectives of TLR including TLD.

- As a result of TLD lacking behind other similar object detection areas, e.g., pedestrian detection and traffic sign detection, we investigated and applied the state of the art learning-based methods ACF and YOLO bringing TLD up to par with the aforementioned similar object detection areas.

- We propose the Bonus Input Layer Single-Shot Multibox Detector (BILSSD) neural network that is an addition to SSD, which besides a regular RGB input image can take one extra input layer, i.e., a foreground probability map, for performing object detection.

REFERENCES

[1] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, July 2016.

[2] M. Jensen, M. Ahrnbom, M. Kruithof, K. Åström, M. Nilsson, H. Ardö, A. Laureshyn, C. Johnsson, and T. Moeslund, "A framework for automated analysis of surrogate measures of safety from video using deep learning techniques." Transportation Research Board National Cooperative Highway Research Program, 9 2018.

[3] A. Møgelmose, M. Trivedi, and T. Moeslund, "Vision based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *I E E E Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 12 2012.

[4] California Department of Transportation. (2015) California manual on uniform traffic control devices. [Online]. Available: http://www.dot.ca.gov/hq/traffops/engineering/control-devices/trafficmanual-current.htm

[5] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Ongoing work on traffic lights: Detection and evaluation," *12th IEEE Advanced Video and Signal-based Survaeillance Conference*, 2015.

[6] M. P. Philipsen and M. B. Jensen, "Computer Vision at Intersections: Explorations in Driver Assistance Systems and Data Reduction for Naturalistic Driving Studies," Master's thesis, Aalborg University, Denmark, June 2015, https://projekter.aau.dk/projekter/files/213565236/main.pdf.

[7] C.-C. Chiang, M.-C. Ho, H.-S. Liao, A. Pratama, and W.-C. Syu, "Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts," *International Journal of Innovative Computing, Information and Control*, vol. 7, pp. 6919–6934, 2011.

[8] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich, "Making bertha see," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 214–221.

[9] R. de Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 358–363.

[10] R. Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 333–338.

[11] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Proceedings of ICRA 2011*, 2011.

[12] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, "Night-time traffic light detection based on svm with geometric moment features," *World Academy of Science, Engineering and Technology 76th*, pp. 571–574, 2013.

[13] M. Ahrnbom, M. B. Jensen, K. Åström, M. Nilsson, H. Ardö, and T. Moeslund, "Improving a real-time object detector with compact temporal information," in *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 190–197.

[14] M. S. Shirazi and B. T. Morris, "Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2017.

[15] World Health Organization. (2017) Save lives - a road safety technical package. [Online]. Available: http://apps.who.int/iris/bitstream/handle/10665/255199/9789241511704-eng.pdf;jsessionid=D2E56E146C7A2A7C73B4D8A67663563F?sequence=1

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[17] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[18] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, June 2011, pp. 3153–3160.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[20] Robotics Centre of Mines ParisTech. (2010) Traffic lights recognition (tlr) public benchmarks. [Online]. Available: http://www.lara.prd.fr/benchmarks/trafficlightsrecognition

[21] H.-K. Kim, J. H. Park, and H.-Y. Jung, "Effective traffic lights recognition method for real time driving assistance system in the daytime," *World Academy of Science, Engineering and Technology 59th*, 2011.

[22] M. Diaz-Cabrera, P. Cerri, and J. Sanchez-Medina, "Suspended traffic lights detection and distance estimation using color features," in *15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1315–1320.

[23] C. Jang, C. Kim, D. Kim, M. Lee, and M. Sunwoo, "Multiple exposure images based traffic light recognition," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1313–1318.

[24] S. Sooksatra and T. Kondo, "Red traffic light detection using fast radial symmetry transform," in *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2014, pp. 1–6.

[25] D. Nienhuser, M. Drescher, and J. Zollner, "Visual state estimation of traffic lights using hidden markov models," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1705–1710.

[26] M. Omachi and S. Omachi, "Traffic light detection with color and edge information," in *2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 284–287.

[27] J. Gong, Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen, "The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 431–435.

[28] C. Wang, T. Jin, M. Yang, and B. Wang, "Robust and real-time traffic lights recognition in complex urban environments," *International Journal of Computational Intelligence Systems*, vol. 4, no. 6, pp. 1383–1390, 2011.

[29] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 707–723, 2012.

[30] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 49–53.

[31] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 809–812.

[32] G. Siogkas, E. Skodras, and E. Dermatas, "Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information." in *VISAPP (1)*, 2012, pp. 620–627.

[33] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[34] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[35] G. Monteiro, P. Peixoto, and U. Nunes, "Vision-based pedestrian detection using haar-like features," *Robotica*, vol. 24, pp. 46–50, 2006.

[36] S. Han, Y. Han, and H. Hahn, "Vehicle detection method using haar-like feature on real time system," *World Academy of Science, Engineering and Technology*, vol. 59, pp. 455–459, 2009.

[37] M. S. Shirazi and B. T. Morris, "Vision-based pedestrian behavior analysis at intersections," *Journal of Electronic Imaging*, vol. 25, no. 5, p. 051203, 2016.

[38] M. S. Shirazi and B. Morris, "Contextual combination of appearance and motion for intersection videos with vehicles and pedestrians," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. McMahan, J. Jerald, H. Zhang, S. M. Drucker, C. Kambhamettu, M. El Choubassi, Z. Deng, and M. Carlson, Eds. Cham: Springer International Publishing, 2014, pp. 708–717.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[40] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[41] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A two-stage approach to people and vehicle detection with hog-based svm," in

Performance Metrics for Intelligent Systems 2006 Workshop, 2006, pp. 133–140.

[42] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study," *Machine vision and applications*, vol. 25, no. 3, pp. 599–611, 2014.

[43] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection at night: Comparison of a learning-based detector and three model-based detectors," *11th Symposium on Visual Computing*, 2015.

[44] M. B. Jensen, M. P. Philipsen, T. B. Moeslund, and M. Trivedi, "Comprehensive parameter sweep for learning-based detector on traffic lights," in *International Symposium on Visual Computing*. Springer, Cham, 2016, pp. 92–100.

[45] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," *18th IEEE Intelligent Transportation Systems Conference*, 2015.

[46] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 5784–5791.

[47] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, "A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles," in *33rd Chinese Control Conference (CCC)*, 2014, pp. 4924–4929.

[48] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[49] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[51] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[52] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[53] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

[54] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[55] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[56] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[57] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[58] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[59] S. Hsu, C. Huang, and C. Chuang, "Vehicle detection using simplified fast r-cnn," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan 2018, pp. 1–3.

[60] Q. Fan, L. Brown, and J. Smith, "A closer look at faster r-cnn for vehicle detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 124–129.

[61] Y. Zhang, J. Wang, and X. Yang, "Real-time vehicle detection and tracking in video based on faster r-cnn," *Journal of Physics: Conference Series*, vol. 887, no. 1, p. 012068, 2017. [Online]. Available: http://stacks.iop.org/1742-6596/887/i=1/a=012068

[62] K. Chen, T. D. Shou, J. K. Li, and C. Tsai, "Vehicles detection on expressway via deep learning: Single shot multibox object detector," in *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 2, July 2018, pp. 467–473.

[63] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, "An improved yolov2 for vehicle detection," *Sensors*, vol. 18, no. 12, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/12/4272

[64] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 443–457.

[65] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, April 2018.

[66] M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating state-of-the-art object detector on challenging traffic light data," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 882–888.

Chapter 4

Semantic

The third main topic of this Ph.D. thesis is Semantic, which is the last stage within the computer vision pipeline seen in Figure 4.1. This mean that the input to the Semantic stage is some trajectories of the detected objects of interest. The task of the Semantic stage is mainly to translate the trajectories of the objects into some high-level understanding, e.g., specific behaviors or abnormal events, with respect to a specific application. In line with the general theme within this Ph.D. thesis of making vehicles able to interpret its surroundings and acquiring video data for automated traffic analysis. The outcome of the traffic analysis is often measures that are defined by traffic researchers rather than computer vision researchers. Working together in cross-disciplinary applications can be challenging as there are different traditions often causing communication challenges and misunderstanding. In this chapter, we will look into frameworks for carrying out automated traffic analysis. Specifically, we will investigate frameworks that allow for cross-disciplinary collaboration for developing traffic analysis tools allowing computer vision based traffic analysis.

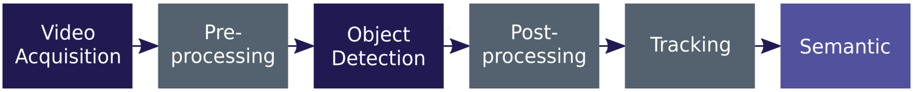


Fig. 4.1: Semantic is the third main topic within this Ph.D. thesis and is the last stage in the overall computer vision pipeline.

The content of this chapter are inspired and have abbreviated parts from [1, 2].

1 INTRODUCTION

Developing computer vision based traffic analysis systems have a considerable interest to computer vision practitioners, in particular, if the system have some research related problem attached to it. The main scope of the computer vision system is to translate the low-level information and mid-level information into some high-level behavior which is often easier to understand even for non-practitioners of computer vision. The low-level information, mid-level information, and high-level information can be described as illustrated in Figure 4.2, which is inspired by [2–5].

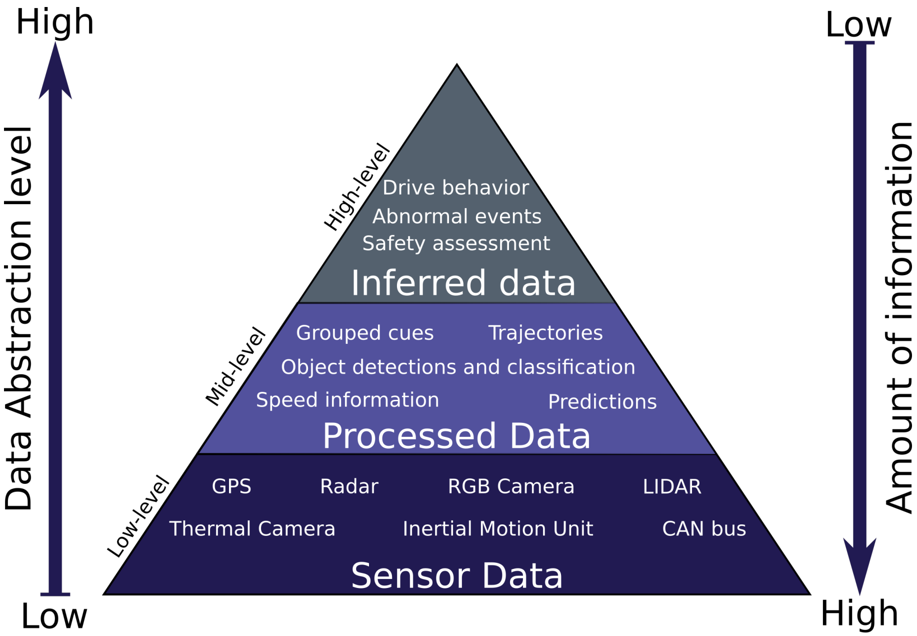


Fig. 4.2: Description of low-level information, mid-level information, and high-level information in respect to computer vision based traffic analysis. The figure is inspired by [2, 3].

The purpose of utilizing computer vision is to automate the process of generating high-level information such as abnormal events, conflict event, object counts, and behaviors, is that it is more efficient compared to traditional approaches. Traditionally, the task of counting the vehicles at an intersection have been carried out by a manual annotator standing physically near the site and marking down each vehicle he or she sees. Other approaches include using video recording equipment to acquire the footage and then let a person look through the acquired data in an offline setup indoor. This is, however, not very efficient, especially not if we are looking for the causes of accidents. By examining the safety pyramid in Figure 4.3, we can - luckily - conclude that accidents do not occur that frequently. The frequency of events are related to the level of its severity in the sense that very severe accidents only occur very seldom, but near-accidents and more normal encounters occur more frequently.

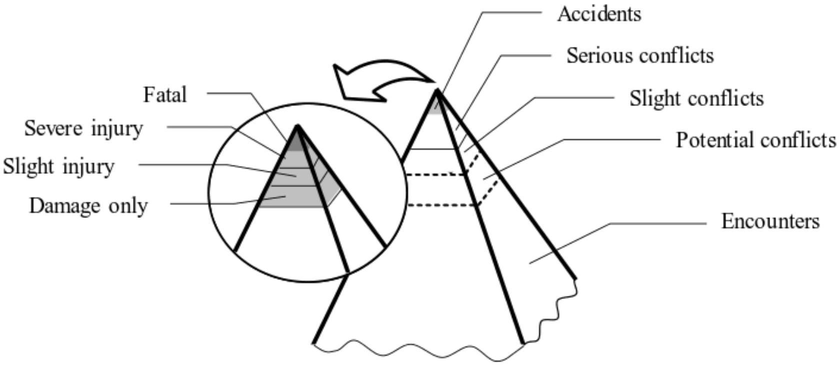


Fig. 4.3: Safety pyramid adopted from [6] and also used in [1].

In order to prevent traffic accidents completely, a lot of data are required in order actually to capture the severe events and thus prevent them. Especially because a significant amount of conflicts and collisions must be recorded before any counteractions can be carried out. Making a person manually annotate the intervals of interests within these enormous datasets is a quite tedious task. The scope of applying a computer vision system to aid this task is thus to automate this tedious task completely. This relationship is seen in Figure 4.4, where the two approaches are plotted at each end of the axis. In between making a human annotate the entire dataset manually and creating a fully automatic computer vision system is the Human-in-the-loop system.

Further, developing a computer vision system that is general enough to work on data captured in Denmark, Sweden, Belgium, Spain, Canada, or the USA, without having to optimize the system each time. As previously mentioned, developing the computer vision system may have some research and development interest from computer vision researchers, but maintaining and optimizing the system and its methods to hundreds of specific traffic intersections across the world may serve a motivational challenge. In these large-scale studies where it is hard to develop a solution that work perfectly, it is often necessary to create the system such humans can adjusted them. This type of system can be referred to as the aforementioned Human-in-the-loop system as seen in both Figure 4.4. The human-in-the-loop aids the computer vision system by adjusting the method's parameters towards the sweet spot

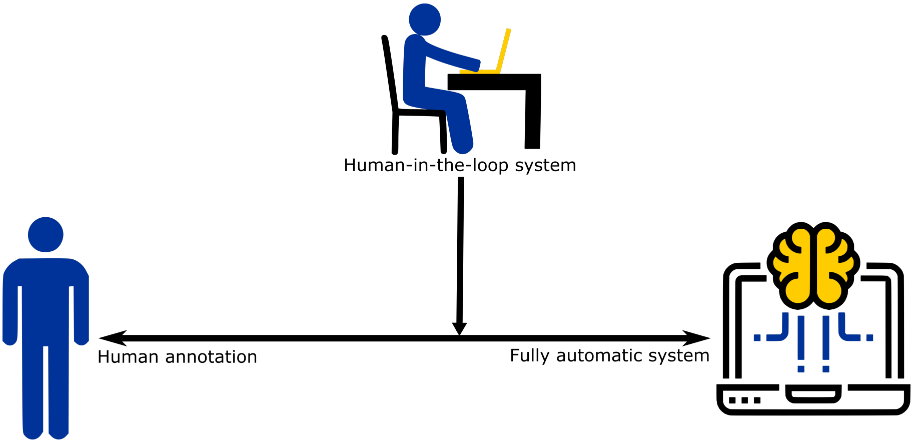


Fig. 4.4: Ideally a fully automatic system should solve the otherwise manual task completely. However, this is often difficult, why a human-in-the-loop system is interesting as it can work as a semi-automated tool.

between finding only relevant events.

To enable non-practitioners of computer vision to become the person that can alter and adjust the computer vision system, the system must be wrapped into a nice package making it accessible for them. This requires different graphical initiatives such as a GUI which is of course important. However, for the system really to be useful, the underlying computer vision methods and its corresponding parameters must have an outlet through the GUI, where the outlet can be adjusted by some measure that a non-practitioner can related to, e.g., direction of movement, velocity, or presence in an area. The task of the human involved in the system is thus to adjust the parameters through the outlet in the GUI, and find the most optimal parameters for the given traffic intersection. This work flow, as well as the trade-offs, are illustrated in Figure 4.5.

There exist quite a few high-level semantic informations that all are of interest for traffic researchers, but it might be with different purposes. In [4] they are interested in safety analysis with respect to estimating the time-to-collision and post-encroachment time which serves as two important safety measurements as they often describe a lot of the near-accidents. Other interesting information can be traffic flow, queue length and congestion detection; road user counts, abnormal behaviors and trajectories; vehicle type classification, speed profiling, object interactions, behavior classification, speed profiling, and other metrics that can be of interest to aid traffic management, driver behavior analysis, and accidents analysis [1, 5, 7–9].

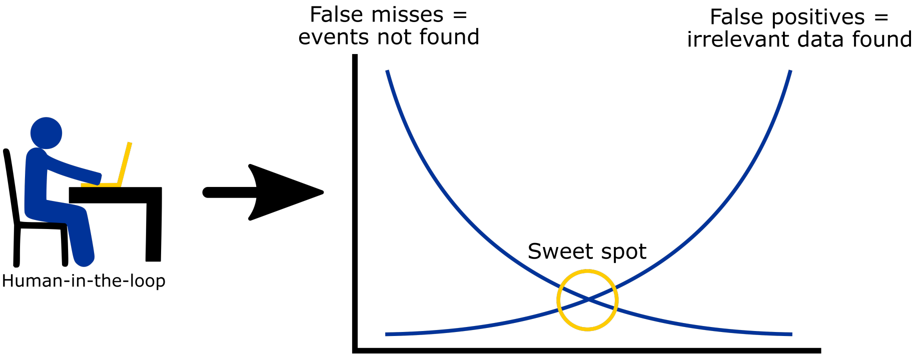


Fig. 4.5: A person can aid the computer vision system in finding the performance sweet spot by adjusting the methods in a human-in-the-loop setup.

2 STATE OF THE ART

The field of automating traffic analysis using computer vision methods has as the introduction suggests been an active research area for a while. This section provides an introduction to popular frameworks which exists with the purpose of aiding this cross-disciplinary discipline. For further introduction, there exists several well-written and thorough surveys on traffic analysis or safety analysis in [4, 7, 10, 11].

Conducting traffic safety studies based on different traffic related metrics and developing computer vision are to some extent two different worlds. This often causes communication challenges between the two research fields when discussing and developing how to come from Video Acquisition to automated high-level Semantic. To ease this communication, various frameworks have been developed and proposed on different technicality levels. [12] presents a general framework in respect to video processing and is to a very large extent applicable for most computer vision applications, especially prior to the introduction of deep learning. The introduced framework consists of six different stages namely: camera, image acquisition, pre-processing, segmentation, representation, and classification. These six stages can be translated into than given a sensor, e.g., a camera, images are captured which can be used to classify objects and events. [13] presents an automated computer vision system that aims to do conflict analysis with respect to pedestrians. This computer vision system is built upon five basic stages: video preprocessing, feature processing, grouping, high-level object processing, and information extraction. Especially the latter stages are more angled towards extracting high-level semantic information compared to the six components presented in [12], this may make them more useful and understandable for non-practitioners of computer vision, i.e., traffic researchers.

In [10] two different approaches or structures for developing automated traffic analysis tools are introduced, namely a top-down approach and a bottom-up approach. The firste stage in the top-down approach consists of finding the foreground objects for example by the use of frame differentiation [14]. The calculated foreground objects are then grouped to form objects and can be classified [15] using either model-based heuristically determined or learning-based models. The classified objects can then finally be linked together in the spatial-temporal domain constituting the tracking stage [16]. The bottom-up approach takes its starting point a little differently by detecting smaller patches of the objects which can then be grouped to a constitute a complete object. Similarly, these objects can be classified to belong to some class label and be tracked across time.

The current available cross-disciplinary frameworks have however not followed the advancement done in computer vision with deep learning.

Object counting is one of the popular high-level semantics that is of high interest of the traffic researchers. Object counting does primarily consist of localizing and classifying objects, which in the traffic perspective is often vulnerable road users, e.g., pedestrians and cyclists, as well as different vehicle types, e.g., cars, trucks, or buses. To prevent the same object being counted multiple times, the objects are tracked which also aid handling occlusion. From a computer vision perspective, a lot of research has been carried out to address detecting and classifying various objects as we have already discussed in Section 2. The object detection methods previously discussed in Section 2 are heavily used for doing object counting as well as most other behavioral studies. For example in [17, 18] a computer vision system is designed for pedestrian behavior analysis at intersections, the detection is based on both motion cues from a GMM as well as appearances cues from Haar-like features which are contextually fused to improve detection. Further, they use HOG and SVM for classification in [18].

To carry out object counting at a point of interest, several solutions are already on the market and widely. The most traditional one, besides hiring a student to manually do the carry out the object count, is installing inductive loops in the road surface which have proven quite accurate in detecting metal objects, e.g, cars, trucks, and lorries, driving over the installation. The simple versions of inductive loops will only count whether a metal object, i.e., a vehicle has passed over the inductive loop installation in the road surface. More advanced versions of inductive loops can, by the use of the magnetic signatures generated by inductive loops, classify the objects into for example cars, trucks, lorries, and even bicycles [19–23]. Inductive loops are, however, and as previously mentioned, required to be installed in the road surface, which is not ideal. Other applications on the market is DataFromSky, who provide *"Advanced traffic analysis of aerial video data"* [24]. DataFromSky explains their application and their overall concept on their website as follows:

"Aerial Monitoring overcomes the limitations of traditional methods of traffic data collection due to its mobility, complexity, and ability to cover large areas. Airborne sensors provide sufficient amount of data to enable vehicle location and movement monitoring; however, further processing and evaluation of the data often requires tremendous effort. DataFromSky is a new specialized solution for automatic analysis of traffic activity in aerial videos. It brings many new possibilities in the field of traffic analysis by virtue of its fully automatic calculation of a wide range of traffic parameters such as speed, densities and gate counting." [24].

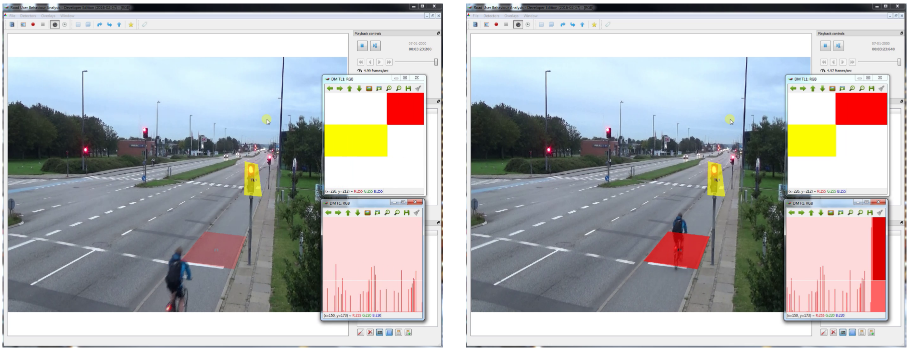
DataFromSky does thus provide semantics high-level information, i.e. advanced traffic analysis, allegedly automatically given a video recording captured from a drone or the likes. However, given it is a commercial product, the precise level of automation is difficult to dissect. For a consumer, however, that might not be of interest at all, as the results DataFromSky produces are excellent. In large-scale studies, this may, however, be a quite expensive solution.

3 CONTRIBUTIONS

During this Ph.D., two works have been carried out regarding Semantics. [1] and [25] do both heavily link to reducing large traffic datasets to only the part of the datasets that are of interest. In [1] (chapter J) we investigate currently available frameworks that aid the communication between traffic researchers and computer vision researchers when developing automated traffic analysis tools. To this extent, we propose a new data-driven framework which aids the cross-disciplinary communications which take its starting point in using modern computer vision technology, i.e., deep learning. Further, we develop a new system, named Surveillance Tracking Using Deep Learning (STRUDL), which is developed and described as per the proposed framework to carry out detection, classification, and tracking which finally can be used to create traffic analysis with data captured from a traffic intersection. STRUDL make use of the data-driven SSD object detector and prior to executing the entire framework, the detector is fine-tuned with a few hundred annotations on a pre-trained model containing popular classes, e.g., pedestrians, cars, and cyclists. To make the most out of STRUDL, a camera calibration is needed of the specific capturing setup with respect to the scene. By combining the re-trained object detector with the camera calibration, trajectories in world coordinates are generated and can easily be extracted to further analysis. Using STRUDL for traffic analysis requires a Linux computer with a newer Nvidia GPU and with CUDA, docker, and nvidia-docker installed on it.

In addition to STRUDL, another, more human-in-the-loop oriented, computer vision watchdog tool is developed and described in [25] (chapter I).

The watchdog tool is called Road User Behaviour Analysis (RUBA). RUBA is developed with a very user-friendly GUI wherein several computer vision method outlets are made available for non-practitioners of computer vision. To this end, the users of RUBA can easily adjust the underlying computer vision methods towards the sweet spot, as illustrated in Figure 4.5, such the most optimal traffic analysis is reached. Compared to STRUDL, RUBA is based on more simple and traditional methods, e.g., background subtraction and optical flow, which in total provide the users of RUBA with four different high-level detectors in RUBA, namely: Presence Detector, Movement Detector, Stationary Detector, and Traffic Light Detector. The detectors do not work on the entire scene captured in the video recordings, the detectors must be annotated in the video recordings manually by the user, as illustrated in Figure 4.6. In Figure 4.6a, a so-called double module is annotated which consists of a Traffic Light Detector, marked with yellow, and a Movement Detector, marked with red. In Figure 4.6a, the Traffic Light Detector is already triggered, which is illustrated by the strong yellow color, whereas the Movement Detector is not yet triggered, which is illustrated with the transparent red color. An event using this double module is triggered the moment both modules are active. In Figure 4.6b, the cyclists runs a red light and cause both modules to be active ultimately triggering an event.



(a)

(b)

Fig. 4.6: .

The contributions made during this Ph.D. thesis to RUBA has mainly been implementing the Traffic Light Detector module allowing for example red running traffic analysis.

The main contributions of this Ph.D. thesis in the field of Semantic can be summarized as follows:

- We propose a new data-driven automated traffic analysis framework that aims towards easing cross-disciplinary development between traffic researchers and computer vision researchers.

- Implementation of the proposed framework that utilizes modern computer vision technology, i.e., deep learning, to carry out conflict analysis in the traffic.

- Developed a Traffic Light Detector module for the RUBA watchdog tool, allowing automated red running analysis or the likes for traffic researchers.

- Both STRUDL and RUBA are open-source projects and are thus made public available at `https://github.com/ahrnbom/strudl` and `https://bitbucket.org/aauvap/ruba/`, respectively.

REFERENCES

[1] M. B. Jensen, M. Ahrnbom, M. Kruithof, K. Åström, M. Nilsson, H. Ardö, A. Laureshyn, C. Johnsson, and T. B. Moeslund, "A framework for automated traffic safety analysis from video using modern computer vision," in *Transportation Research Board (TRB) 98th Annual Meeting*, 2018.

[2] M. P. Philipsen, M. B. Jensen, R. K. Satzoda, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, "Day and night-time drive analysis using stereo vision for naturalistic driving studies," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 1226–1231.

[3] R. K. Satzoda and M. M. Trivedi, "Drive analysis using vehicle dynamics and vision-based lane semantics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 9–18, 2015.

[4] M. S. Shirazi and B. T. Morris, "Investigation of safety analysis methods using computer vision techniques," *Journal of Electronic Imaging*, vol. 26, no. 5, p. 051404, 2017.

[5] M. S. Shirazi and B. T. Morris, "Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2017.

[6] C. Hydén, "The development of a method for traffic safety evaluation: the swedish traffic conflict technique," *"Department of Traffic Planning and Engineering. Bulletin 70"*, vol. Doctoral thesis. Lund University, 1987.

[7] B. T. Morris and M. M. Trivedi, “A survey of vision-based trajectory learning and analysis for surveillance,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, Aug 2008.

[8] B. T. Morris and M. M. Trivedi, “Learning, modeling, and classification of vehicle track patterns from live video,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, Sept 2008.

[9] B. Morris and M. Trivedi, “Robust classification and tracking of vehicles in traffic video streams,” in *2006 IEEE Intelligent Transportation Systems Conference*, Sept 2006, pp. 1078–1083.

[10] N. Buch, S. A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, Sept 2011.

[11] B. T. Morris and M. Trivedi, “Understanding vehicular traffic behavior from video: a survey of unsupervised approaches,” *Journal of Electronic Imaging*, vol. 22, no. 4, p. 041113, 2013.

[12] T. Moeslund, *Introduction to video and image processing: Building real systems and applications*. Springer, 2012.

[13] K. Ismail, T. Sayed, N. Saunier, and C. Lim, “Automated analysis of pedestrian-vehicle conflicts using video data,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2140, pp. 44–54, 2009.

[14] S.-C. S. Cheung and C. Kamath, “Robust background subtraction with foreground validation for urban traffic video,” *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 14, p. 726261, 2005.

[15] N. Buch, J. Orwell, and S. A. Velastin, “Urban road user detection and classification using 3d wire frame models,” *IET Computer Vision*, vol. 4, no. 2, pp. 105–116, 2010.

[16] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, “Coupled object detection and tracking from static cameras and moving vehicles,” *IEEE PAMI*, vol. 30, no. 10, pp. 1683–1698, 2008.

[17] M. S. Shirazi and B. T. Morris, “Vision-based pedestrian behavior analysis at intersections,” *Journal of Electronic Imaging*, vol. 25, no. 5, p. 051203, 2016.

[18] M. S. Shirazi and B. Morris, “Contextual combination of appearance and motion for intersection videos with vehicles and pedestrians,” in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin,

R. McMahan, J. Jerald, H. Zhang, S. M. Drucker, C. Kambhamettu, M. El Choubassi, Z. Deng, and M. Carlson, Eds. Cham: Springer International Publishing, 2014, pp. 708–717.

[19] K. Nordback, D. P. Piatkowski, B. N. Janson, W. E. Marshall, K. J. Krizek, and D. S. Main, "Using inductive loops to count bicycles in mixed traffic," *Journal of Transportation of the Institute of Transportation Engineers*, 2011.

[20] J. Gajda, P. Piwowar, R. Sroka, M. Stencel, and T. Zeglen, "Application of inductive loops as wheel detectors," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 57–66, 2012.

[21] H. A. Oliveira, F. R. Barbosa, O. M. Almeida, and A. P. Braga, "A vehicle classification based on inductive loop detectors using artificial neural networks," in *Industry Applications (INDUSCON), 2010 9th IEEE/IAS International Conference on*. IEEE, 2010, pp. 1–6.

[22] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, "A vehicle classification based on inductive loop detectors," in *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, vol. 1. IEEE, 2001, pp. 460–464.

[23] Y.-K. Ki and D.-K. Baik, "Vehicle-classification algorithm for single-loop detectors using neural networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 6, pp. 1704–1711, 2006.

[24] DataFromSky. (2018) Datafromsky website @ONLINE. [Online]. Available: http://datafromsky.com/

[25] Bahnsen, Chris H. and Madsen, Tanja K. O. and Jensen, Morten B and Lahrmann, Harry S. and Moeslund, Thomas B., *The RUBA Watchdog Video Analysis Tool*. Deliverable submitted within the InDeV EU project. Based on the public wiki page., 2018. [Online]. Available: https://bitbucket.org/aauvap/ruba/wiki/

Chapter 5

Knowledge for the World

The modern technology can be applied to a large variety of applications positively benefiting the specific area. To increase the chances of deploying modern technology in new areas, the public need to be aware of the possibilities, in particular, the decision makers. Translating the advancement in computer vision to layman's terms and then disseminating them to the public is what this last topic covered in this Ph.D. thesis is about. The topic is titled after the overall strategy of Aalborg University (AAU) [1].

1 INTRODUCTION

One of the fundamentals in the entire backbone of AAU is that everything we do is done in accordance with a problem-based approach. The research, as well as our educations, are built upon problem-based learning. This is well supported by the future direction and future course of the university described in the AAU Strategy 2016-2021 [1], which is, as previously mentioned, titled Knowledge For the World. To accommodate the future direction, we as researchers as well as educators are always encouraged to pick up research and problems both globally and nationally, but in particular in our local communities which will eventually strengthen the province. Further, the research and problems we address must be linked to a real-world problem preferably in close collaboration with external partners in the business world or in public institutions [1].

The overall characteristics and goals at AAU are well-defined in the AAU Strategy 2016-2021 [1] which are as follows:

- *Problem orientation*

> *AAU's problem based approach to research and education is strong and*

> *well founded. Our researchers, students and graduates are well trained in analytical, holistic and problem and solution oriented methods.*

- *Collaboration*

  > *AAU conducts research in close collaboration between staff, students and partners in the business world and in public institutions. Working with authentic issues implies that the University maintains close contact with external partners.*

- *Commitment*

  > *AAU reflects the vigour and zeal of its staff and students. AAU is a university for committed staff and students who assume responsibility and make things happen within the University and in its surrounding society.*

- *Change*

  > *AAU creates knowledge that changes the world. Our problem oriented approach to research, education, knowledge dissemination and collaboration makes a difference and create change.*

These above characteristics, which to some extent are metrics in AAU regime, do in respect to this Ph.D. thesis not constitute any contributions to the technical world of computer vision. However, that is not the purpose either. The purpose of doing this is first of all to disseminate the research you work on, which is an essential skill to learn and eventually master. Mainly because people possess very different educational levels and general understanding of what is going on the specific topic. Especially a topic such as computer vision which is regularly being thrown into the buzzwords mix of artificial intelligence, self-driving vehicles, and the always interesting sci-fi doomsday theory, i.e., robots taking over the world. It is therefore essential to keep educating people by explaining how research is conducted, how does it work, how does it perform, and how should it eventually be used - and where should it not be used. Publishing popular science papers put a focus on the advancement done in a field of study, which may attract young scholars to pursue a career in the specific field. Engaging positively in relevant discussions and debates will eventually improve and be decisive for the common public sentiment as people by nature are more reluctant with things they do not entirely understand. So if we eventually want to convince people to put their faith in a self-driving vehicles, we need to make sure they understand how it works on a high-level. The low-level understanding remains in the hands - or brains - of researchers.

2 CONTRIBUTIONS

In addition to various teaching and supervision tasks, two initiatives have been carried out during this Ph.D. in respect to this Knowledge for the World topic. Both of them have been to translate the latest advancement in computer vision to the public through two different channels. The main overall advancement done within computer vision during this Ph.D. period has been deep learning. To create more interest about computer vision and deep learning among high school students, we wrote a popular scientific paper about deep learning to *Aktuel Naturvidenskab* in 2018 [2] (chapter K). Aktuel Naturvidenskab is a Danish popular science magazine dealing with current science subjects and is published in collaboration between six Danish universities. Aktuel Naturvidenskab does to some extent correspond to a Danish version of the American popular science magazine called Scientific American. The magazine is, however, published in Danish and distributed to most of the Danish high schools, why the technical level of the paper has been adjusted accordingly. The publications in Aktuel Naturvidenskab are not peer-reviewed, and do thus not constitute a scientific contribution to the world of computer vision. However, our publication on deep learning has spawned several interviews with high schools students as well as students from primary school who has been intrigued about learning more about deep learning, the perspectives, and how to get started.

To reach a broader and more varied group, we developed a Techtunnel showcasing the latest advancement within computer vision, their corresponding applications, and the future perspective. We developed the Techtunnel specifically with the purpose of participating in the Danish Folkemøde 2018, which can be translated into The People's Meeting. The People's Meeting is a yearly recurrent four days event with an estimated 113,000 attendees in 2018. The development and outcome of this work been documented in a technical report used internally at Aalborg University [3] (chapter L). During the four days, around 25,000 people passed through the Techtunnel, and a large portion of them engaged in friendly conversions and discussions about computer vision with us. Further, the Techtunnel at The People's Meeting spawned a lot of media attention which led to two TV interviews with the Ph.D. student on the local news channel TV2 Bornholm as well as the national news channel TV2 News, respectively.

The main contributions of this Ph.D. thesis within the field of Knowledge for the World can thus be summarized as follows:

- We have published a popular science paper on the advancement in deep learning which has been distributed among most Danish high schools spawning much interest in the topic.

- We showcased the latest advancements within computer vision and their many applications in the Techtunnel at The People's Meeting. Our presence spawned two TV interviews on local and national channels, respectively.

REFERENCES

[1] Aalborg University, *Knowledge for the World - Aalborg University Strategy 2016-2021*. Aalborg University, 2015. [Online]. Available: http://www.e-pages.dk/aalborguniversitet/383/

[2] M. B. Jensen, C. H. Bahnsen, K. Nasrollahi, and T. B. Moeslund, "Deep learning: Et gennembrud inden for kunstig intelligens," *Aktuel Naturvidenskab*, vol. 2, 2018.

[3] M. B. Jensen, M. Pedersen, and P. Lund, "Techtunnel - folkemødet 2018," *AAU Technical Report*, 2018.

Chapter 6

Conclusion

Running from 2015-2018, this Ph.D. has covered topics in driving assistance systems and looking at traffic analysis from a computer vision perspective. The research within the two topics are carried out in three different stages in a general computer vision pipeline. The three computer vision stages investigated in respect to driving assistance systems and looking at traffic analysis are marked with dark blue in Figure 6.1.

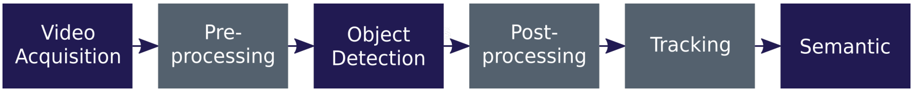


Fig. 6.1: The computer vision stages involved in this thesis are marked in blue. In addition to the stages marked in dark blue, this Ph.D. thesis also include a Knowledge for the World topic.

In addition to the stages marked in dark blue, this Ph.D. thesis also include a Knowledge for the World topic. The four main topics investigated within this Ph.D. thesis are thus:

- Video Acquisition
- Object Detection
- Semantic
- Knowledge for the World

Within the main topic of Video Acquisition, a multi-modal and multi-view recording setup is developed to accommodate the requirements introduced by the InDeV project around in Europe. The setup has successfully been used to capture several months of video data within the InDeV project. The development of the recording setup indicated that the use of portable pole may provide less occlusion-prone video recordings, which ultimately improves the following traffic analysis. To this end, we present a comprehensive survey and overview of available portable poles, both complete solutions and do-it-your-own solutions. The survey and overview point out the lack of lightweight, robust, and yet mobile portable pole. To accommodate these requirements, we design and develop a new portable pole satisfying these requirements. The new portable pole has already seen its use in many traffic analysis studies carried out by the Traffic Safety Research Group at Aalborg University. The portable pole has provided useful traffic video recordings in rural areas which were no, or a very limited, amount of existing infrastructure was available. Further, the portable pole has seen its great use in urban environments as well as it do not occupy much of the pavement. Though the new portable pole provides a larger degree of freedom of the exact capturing position, it does not eliminate occlusion. To investigate the occlusion issue further, we collect a synchronized dataset from a traffic intersection from respectively an existing infrastructure view-angle and a drone view-angle. The drone is occasionally argued as the ideal countermeasure for occlusion, but as the preliminary research carried out with this thesis show, it does indeed serve as an excellent capturing view-angle when detecting larger objects such as cars, trucks, and buses. However, the drone view-angle is challenged a lot by smaller objects in the scene such as pedestrians, cyclists, and mopeds. Going forward there are still many investigations required to actually determine whether drones are superior as a capturing angle compared to existing infrastructure or portable poles. But when are they exactly superior - if they are. The assembled dataset allow for these further investigations on both a bounding boxes level but also on a pixel level.

In the area of Object Detection, a comprehensive and thorough traffic light survey was made to provide an overview of the work done with respect to recognizing traffic lights. The output of the survey clearly showed that the area of traffic light recognition including detection had not seen the same usage of learning-based methods as related areas had seen, e.g., pedestrian detection and traffic sign detection. Further, the survey showed a lack of a large and challenging traffic light dataset with a well-defined evaluation process, which have made the comparison of the previous research difficult. To address this need, the LISA Traffic Light Dataset was collected, annotated, and made freely available with the survey to accommodate fair comparison of methods within the area of traffic light detection. To bring traffic light detection up to par with the aforementioned related areas, we investigated how to apply and develop the modern computer vision technology to push the state of the art detection performance. The usage of learning-based method ACF and in particular YOLO did quickly outperform the previously heuristically determined models. The research carried out during this Ph.D. in respect to traffic light detection has brought the state up to par with the related areas and has to some extent solved the daytime challenge within the LISA Traffic Light Dataset. Looking ahead in the field of traffic light recognition in regard to driving assistance systems and autonomous vehicles, more varied and challenging data are required. Especially datasets from more challenging environments than sunny San Diego, California are of high importance. Further, publicly available datasets where the annotated images also contain accurate GPS coordinates, will eventually aid the real-life performance of traffic light detection. These kinds of datasets will potentially help to solve one of the biggest challenges of both traffic light detection but also traffic sign detection, which is to determine the relation and importance of traffic light in respect to the vehicle and its location on the road.

For Semantic, a new data-driven framework for easing the cross-disciplinary task of developing an automated traffic analysis tool is introduced. To this extent, an automatic and a semi-automatic computer vision-based traffic analysis tool have been developed and made open-source and freely available for traffic researchers across the world. The semi-automatic tool, RUBA, has been developed in close collaboration with the Traffic Research Group (TRG) at Aalborg University, such several traditional computer vision methods have been made available for non-practitioners of computer vision. RUBA is a human-in-the-loop computer vision system, which due to the close collaboration with TRG has resulted in a very user-friendly GUI interface. RUBA has seen its use in the entire world, reports and feedback shows that, besides being used by the InDeV partners in Europe, several other both public and private actors has used RUBA, among others are:

- Aalborg Municipality, Denmark
- ATKI, Denmark
- NTNU, Norway
- Oregon State University, USA
- Transport Research Centre, Czech Republic
- Transportøkonomisk Institutt, Norway
- Indian Institute of Technology Bombay, India
- National University of Singapore, Singapore
- Tallinn University of Technology, Estonia
- University of Coimbra, Portugal

The contributions made during this Ph.D. thesis to RUBA has mainly been implementing the Traffic Light Detector module allowing for example red running traffic analysis.

Future work in this area includes making deep learning methods even more accessible for traffic researchers as well as other field of research.

In the area of Knowledge for the World, two main initiatives have been carried out. A popular science paper about deep learning has been published in a Danish magazine and distributed to most Danish high schools and some primary schools. The publications in Aktuel Naturvidenskab are not peer-reviewed, and do thus not constitute a a scientific contribution to the world of computer vision. However, our publication on deep learning has spawned several interviews with high schools students as well as students from primary school who has been intrigued about learning more about deep learning, the perspectives, and how to get started. To reach an even broader and more varied group, we developed the Techtunnel and brought it to the Danish Folkemøde, "The People's Meeting," where more than 110,000 people attended the entire four-day event. Around 25,000 people visited our Techtunnel where we showcased the latest advancement within computer vision and the current as well as future applications the new technologies create. The Techtunnel resulted in quite some media attention, which led to two TV interviews on the local and the national TV channel, respectively. Looking forward in respect to Knowledge for the World, it remains an important task to disseminate the advancements being done within a researcher's field of study in layman terms to the public. Further, we need to keep showcasing the many and in particular cool opportunities within science to our younger generations, such we can keep attracting more young people into science.

The main contributions of this entire Ph.D. thesis can be summarized as follows:

Video Acquisition

- Developed a multi-view and multi-modal recording platform for video acquisition at traffic intersection. The recording platform has captured several months of data both in Denmark and Belgium.
- We have surveyed and provides an overview of four general types of portable poles with respect to capturing data at traffic intersections in rural and urban environments.
- The portable pole overview pointed out the lack of a lightweight, robust yet mobile portable pole. We have thus designed and developed a new portable pole satisfying these requirements.
- We have collected and annotated a synchronized multi-view traffic intersection dataset allowing performance comparison between video acquired with a drone and video acquired by installing camera equipment in existing infrastructure, respectively.
- Preliminary result based on a pre-trained Mask R-CNN object detector shows that the drone view-angle is superior to the existing infrastructure view-angle when looking for larger objects, e.g., cars, trucks,

and lorries, but when looking at pedestrians and cyclists, the roles are flipped as none of these are detected in the drone view-angle.

Object Detection

- The public available LISA Traffic Light Dataset which consists of 43,007 frames and 119,231 annotations split between 304 unique traffic lights captured during both daytime and nighttime.
- In addition to the dataset, we have proposed a standardized way of evaluating traffic light detection on the LISA Traffic Light Dataset easing comparison between methods.
- We have provided a comprehensive overview of the current issues, challenges, used methods, and perspectives of traffic light recognition including traffic light detection.
- As a result of traffic light detection lacking behind other similar object detection areas, e.g., pedestrian detection and traffic sign detection, we investigated and applied state of the art learning-based methods ACF and YOLO bringing TLD up to par with aforementioned similar object detection areas.
- We propose the Bonus Input Layer Single-Shot Multibox Detector (BILSSD) neural network that is an addition to SSD, which besides a regular RGB input image can take one extra input layer, i.e., a foreground probability map, for performing object detection.

Semantic

- We propose a new data-driven automated traffic analysis framework that aims towards easing cross-disciplinary research between traffic researchers and computer vision researchers.
- Implementation of the proposed framework that utilizes modern computer vision technology, i.e., deep learning, to carry out conflict analysis in the traffic.
- Developed a traffic light module for the RUBA watchdog tool, allowing automated red running analysis or the likes for traffic researchers.
- Both STUDL and RUBA are open-source projects and are thus made public available at `https://github.com/ahrnbom/strudl` and `https://bitbucket.org/aauvap/ruba/`, respectively.

Knowledge for the World

- We have published a popular science paper on the advancement in deep learning which has been distributed among most Danish high schools spawning a much interest in the topic.
- We showcased the latest advancements within computer vision and their many applications in the Techtunnel at The People's Meeting. Our presence spawned two TV interviews on local and national channels, respectively.

This work has mainly been concerned about traffic analysis and driving assistance systems, there are however so many other applications and research areas that can greatly benefit by computer vision. In particular, cross-disciplinary domains where two research areas can greatly benefit from each other's challenges and solutions. In the perspective of traffic applications, much work still has to be done. Making a vehicle drive entirely autonomous requires so many research problems to be further investigated and developed, additionally the interlink between them and how they compensate each other are essential. In the same way, automated traffic analysis, can to some extent, be automated in a very optimized setup in a well-suited traffic intersection, but if these tools were to be rolled out in a large-scale much work is still required to cope with the large variance. Finally, a vital thing for the entire field of computer vision is that we must keep disseminating things and showcase the exciting research that is done. Partly because we must keep attracting young and bright scholars, but mainly because we need to help the society understand how this new automation adventure works. If they do not understand it on some high-level, they will never trust it completely which may hinder the enormous potential of increasing the usage of computer vision in daily lives even more.

Part II

Video Acquisition

Paper A

InDeV Recording Setup

Morten B. Jensen and Thomas B. Moeslund

This technical report has been used for internal dissemination in the Horizon 2020 EU project: InDeV - In-Depth understanding of accidents causation for Vulnerable road users. The technical report has been written in 2016 and revised in 2018.

ABSTRACT

The scope of the Horizon 2020 EU project: In-Depth understanding of accidents causation for Vulnerable road users (InDeV) is to contribute to the overall improvement of Vulnerable Road Users' (VRU) safety in Europe. Interestingly, road safety experts are concerned about the issue of having "to few crashes" in their observation data to propose an accurate accidents causation. It is therefore desirable to get a much observation data as possible to provide an accurate cause of accidents. But rather than deploying a person to physically monitor an area of interest, it is more ideal to install video recording equipment to observe the area. In this technical report we design and develop a multi-view and multi-modal video recording system to aid the observational data study within the InDeV project. The video recording system is designed to be usable in the two main objectives in the InDeV project, namely: Short-term observations (3-weeks) and long-term observations (1-year). The video recording system is developed with the purposed of being deployed at 24 different traffic intersection spread across Belgium, Denmark, Netherlands, Poland, Spain, and Sweden in both short-term observations as well as long-term observations.

1 INTRODUCTION

Understanding the causes of accidents is an important pre-requisite for developing effective counter-measures and improving safety in general. At this point, the In-depth Understanding of Accident Causation for Vulnerable Road Users (InDeV) project comes into play. The main objective of the project is to develop an integrated methodology to study causes of accidents with vulnerable road users (such as pedestrians, cyclists and moped riders). [1]

Traditionally, traffic conflict studies relied on using human observers who had to detect and judge conflicts in real time while being present in the traffic environment at the studied site. But rather than deploying a person to physically monitor an area of interest, the task is eased by installing video recording equipment in the area instead. The record video can either be examined by a human observer offline or more ideally by the use of computer vision tools, e.g. RUBA [2], STRUDL [3], etc.

To address and explore the main objective of the InDeV project, a video recording system is required to facilitate the extensive amount of recordings desired to do within the capture. A total of 21 short-term recordings and 3 long-term recordings are expected to be carried out during a 3-years period between 2016-2018. The short-term recordings consists of 3-weeks continuous video recordings at 21 different traffic intersections spread across Belgium, Denmark, Netherlands, Poland, Spain, and Sweden. The long-term recordings will be carried out at Belgium, Poland, and Spain by capturing 3 different traffic intersections continuously throughout a period of 1-year each site. All of the video recordings are done with the purpose of ultimately aiding the understanding of accident causation for vulnerable road users (VRUs). [1]

To accommodate these demands, We develop a multi-view and multi-modal video recording system to aid the observational data study within the InDeV project.

2 REQUIREMENT CONSIDERATIONS

Prior to designing and developing the video recording system, a few considerations in relation to the requirements of the recording system is done. These requirement considerations presents some common and essential demands for the solution when deploying a video recording system for a period of up to 3-weeks.

Multi-view and multi-modal To achieve the best possible data for a traffic intersection of interest, the InDeV project will capture data 24 hours a day in a period of 3 weeks. Using only a RGB camera will be applicable during the periods of the day with well lighting, however during night-time, the video feed is very limited. To this extend it would be useful to include thermal cameras as a complimentary sensor to the RGB camera.

A classical issue with most video recording systems later used with computer vision methods is occlusion. It is therefore desirable if the video recording setup allow for a multi-view setup providing video from different angles of the traffic intersection. We propose to use two different view-angles in a normal sized traffic intersection. The first view-angle would make use of a RGB camera as well as a thermal sensor while the second view-angle will only have a RGB camera.

Power supply The most essential demand is to find a power solution that work without a stable power supply and can thus become independent of existing infrastructure. Batteries are the obvious choice to solve this task, but to reach the 3 weeks operations time this might require large set of batteries. Rather than having batteries for all 3 weeks, a smaller set of batteries could be used, changed, and recharged during operations hours. This might however require a person on-site on a daily basis. Ideally, the system should be kept in operational mode during the battery switch requiring battery hot-swapping mechanisms.

Vandalism prevention A major issue with having equipment portable and mobile is that it becomes more prone to vandalism and theft. Prevention of vandalism and theft is very difficult, but the physical setup must take this into consideration.

Recording capacity 3 cameras which are constant recording throughout 3 weeks requires a large amount of storage capacity. This sets some demands of the on-site equipment as the cameras should ideally be kept in operation all the time. A different aspect, in connection to the vandalism prevention point, is, if all recordings are left on-site and not copied or backed-up during the operation time, the recordings could in worst case be completely lost.

Video from the thermal camera at VGA resolution(640x480) demands an estimated 0.245 GB/hour, while the RGB video at full HD resolution(1920x1080) on average requires 6.9 GB/hour. The estimated storage requirement for the continuous operation for 3 weeks with one thermal camera and two RGB cameras is 7,100 GB. If the storage capacity is divided into the physical locations of the cameras, it translates to an estimated 3,600 GB for the RGB+thermal setup and 3,500 GB for the single RGB camera.

3 RECORDING SETUP ANALYSIS

Several initiatives and brainstorms have been made in creating the InDeV video recording setup, all with the purpose of elucidating approaches for collecting data in the traffic scene in the best possible manor. The scope of the analysis has mainly been focused on the short-term recordings (3-weeks).

3.1 Base framework

In the requirement considerations in Section 2 it is defined that the three cameras should be used in a 2 view-angles setup. We further assume that these view-angles are not physically connected to each other, not even by cable.

Most industrial cameras today operates by Power over Ethernet (PoE) which means it is only necessary to supply one cable per camera in the mast. In our recording setup we propose to use is presented in Table A.1. The power is supplied by a separate PoE switch which transmits 19 V for each RJ45 port. AXIS provides its own Media SDK which is available for Windows and allows simultaneous acquisition from multiple network cameras. Furthermore, we need to record the video on one or several external hard drives

Table A.1: Capturing devices the InDeV video recording setup.

Type	*Manufacturer*	*Model*	*Weight[kg]*	*Price[Euros]*
RGB	Axis	Q1615-E	3.5	1,227.13
Thermal	Axis	Q1932-E	2.2	9,000.00

with a total capacity of no less than 7,100 GB. The aforementioned requirements implies that the recording devices should interface with an both hard drives and ethernet. It should either run a x86-compatible version of Windows or be a separate Network Allocated Storage (NAS) server. Due to the limited power available, this means we would either need a low-power Windows device such as a small laptop/netbook, an Intel NUC, or a dedicated NAS server.

The video acquisition should be synchronized across multiple view-angles, even if the view-angles are not physically connected. In order to assure this, we must synchronize the system clocks of the cameras and recording devices against the same server, for instance via the Network Time Protocol (NTP). This in turn requires an internet connection which may be provided by a UMTS/LTE modem. The modem also allows the remote monitoring of the system to check the state of the cameras and power supply.

We summarise the required base setup to support a single pole configuration in the following itemization:

- Power over Ethernet (PoE) Switch.
- Recording solution consisting of either:
  - Low-power x86-compatible Windows device with full-size USB, RJ45 port + External hard drive(s) with combined capacity of at least 7,100 GB.
  - Dedicated NAS-server.
- UMTS/LTE Modem.
- Network cables.
- Power adapters.

An example configuration based on the above itemization is listed in Table A.2. Since the two view-angles are not physically connected, the configuration should be doubled for each site. The total price of the PC-configuration is 660 Euros whereas the NAS-configuration is priced at 605 Euros. If it is possible to share the internet connection by a local WiFi network at the acquisition site, only one UMTS/LTE model is required.

3.2 Power supply and enclosure

The base framework presented above must be powered throughout the 3-weeks acquisition period. The power supply and the base framework must also be placed in an enclosure which is resistant to tampering. We use the example configuration of Table A.2 and the cameras specified in Table A.1 to

Table A.2: Example configuration of the base framework. We assume that the external hard drive is changed before its capacity is exceeded. The power for the external hard drive is supplied through the Windows device.

Device	Manufacturer	Model	Price [Euros]	Power consumption
PoE switch	LevelOne	FSW-0503	125	3 W
UMTS/LTE Modem	TP-Link	TL-MR3040	125	4 W
Windows device	ASUS	Eee PC 1001PXD or similar	230	16 W
External hard drive	Toshiba	2 x StorE Canvio, 2 TB	180	-
NAS server	Synology	DS215j	175	13.4 W
Internal hard drive	WD	2 x Purple WD20PURX 2 TB	180	-

Table A.3: Estimated power consumption of camera setups.

Device	Power consumption	Consumption in three weeks (504 hours)
Example config. of Table A.2 with PC	23 W	11,592 Wh
Example config. of Table A.2 with NAS	20.4 W	10,282 Wh
Axis Q1615-E	3.7 W	1,865 Wh
Axis Q1932-W	6 W	3,024 Wh
Total config (PC) + RGB+thermal	32.7 W	16,481 Wh
Total config (PC) + RGB	26.7 W	13,457 Wh
Total config (NAS) + RGB+thermal	30.1	15,171 Wh
Total config (NAS) + RGB	24.1 W	12,147 Wh

estimate the power consumption of the setup. This estimate is calculated in Table A.3 for the RGB+thermal setup and for the RGB setup alone.

We use the estimated power consumption of Table A.3 to calculate the required size of the batteries and the enclosure where the batteries should reside. The calculations use the NAS configuration as the point of departure. We deal with two scenarios as:

1. The setup should be self-contained and requires no manual intervention

throughout the acquisition period. Thus, the batteries must contain all the power required.

2. The setup might require replacement of batteries throughout the acquisition period. This decreases the number of batteries and thus reduces the size of the enclosure.

The batteries used for the calculations are heavy-duty 12 V batteries, for instance the Biltema SMF 12 V 180 Ah which costs 242 Euros or the Varta M18 12 V 180 Ah at 255 Euros. The dimensions of the batteries are 513x223x223 mm and the weight 45.1 kg. Scenario # 1 requires at least 10 batteries for the RGB+thermal setup and 8 batteries for the RGB setup. We calculate an overhead of 30 % for each setup as the real-world capacity of the batteries is lower than the specifications.

If we use 3 batteries for scenario # 2, the estimated replacement cycle is 6 and 8 days for the RGB+thermal and RGB setup, respectively.

A suitable enclosure for the batteries and associated equipment found in Table A.2 are the Eurobox and Alu-Box manufactured by the Danish company Zarges. The Eurobox has previously been used to store batteries and equipment for another traffic monitoring study performed at AAU, containing 2-3 batteries. This configuration is shown in Figure A.2. The Alu-Box is a rugged, sturdier, and IP65-certified version of the Eurobox. The enclosure options and the number of batteries they may contain is listed in Table A.4.

Fig. A.1: Enclosure options listed by the corresponding number of batteries that fits inside the box.

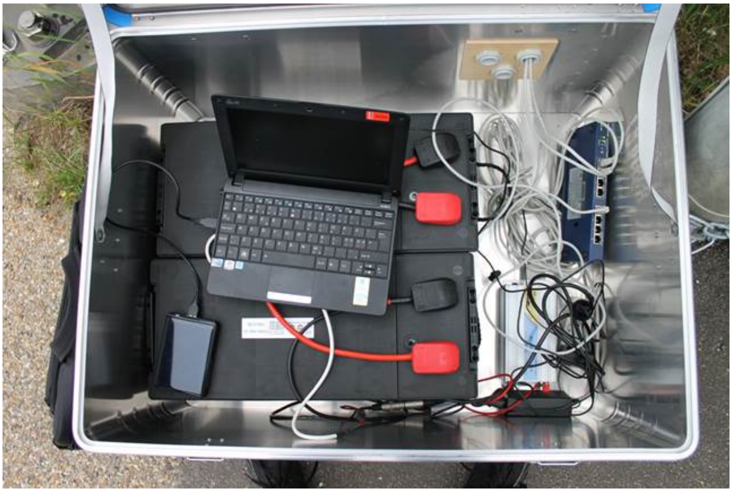


Fig. A.2: The Eurobox 40705 containing two batteries, a 220 V power converter, a PoE switch, and a netbook. The third battery is added by rotating the existing batteries 90 degrees.

Table A.4: The enclosure options and the corresponding possible number of batteries.

Battery capacity	Model	Outer dimensions	Inner dimensions	Capacity	Weight	Price [Euros]
3	Eurobox 40705	800x600x410 mm	750x550x380 mm	155 L	7.5 kg	428
3	Alu-box 40565	800x600x410 mm	750x550x380 mm	157 L	10.0 kg	540
6	Eurobox 40706	800x600x610 mm	750x550x580 mm	240 L	8.9 kg	493
6	Alu-box 40566	800x600x610 mm	750x550x580 mm	239 L	12 kg	609
12	Eurobox 40709	1200x800x500 mm	1150x750x480 mm	400 L	16.9 kg	820
12	Alu-box 40580	1200x800x510 mm	1150x750x480 mm	414 L	20 kg	990

4 VIDEO RECORDING SETUP

The developed video recording setup at a intersection will consists of two setups: the first setup will have one RGB and one thermal camera and the second setup will have one RGB camera. Each setup will be placed strategically in the intersection in order to prevent occlusion. Each setup consists of the same basic recording equipment; A Network Attached Storage (NAS) server, network switch, and batteries. The recording equipment is placed in an aluminium box that is chained to existing infrastructure. Having two setups at one intersection, that are not wired together, presents a time synchronization challenge. This has been solved by modifying one of the NAS servers to run a Network Time Protocol (NTP) server and setting the NAS servers in a master-slave configuration, such that the first NAS runs a NTP server and the second NAS synchronizes towards this one. By including a wireless access point to the recording equipment for each setup, the two setups are wirelessly connected; hence they can synchronize with each other. The layout of the video recording setup is presented in Figure A.3.

The cameras are installed in existing infrastructure by the use of a lift as seen in Figure A.4

The cameras are manually installed and mounted on the traffic light pole, the 2 different view-angles are installed diagonally across the traffic intersection as seen in Figure A.5

4.1 Time synchronization issues

The setup is created such that all 3 cameras are time-synching towards one of the NAS in the setup(The NAS where 2 cameras are connected physically). It uses the NTP protocol to adjust the timedrift. The shortest adjustment interval is 16 seconds, but as it is not a static interval, it changes as a results

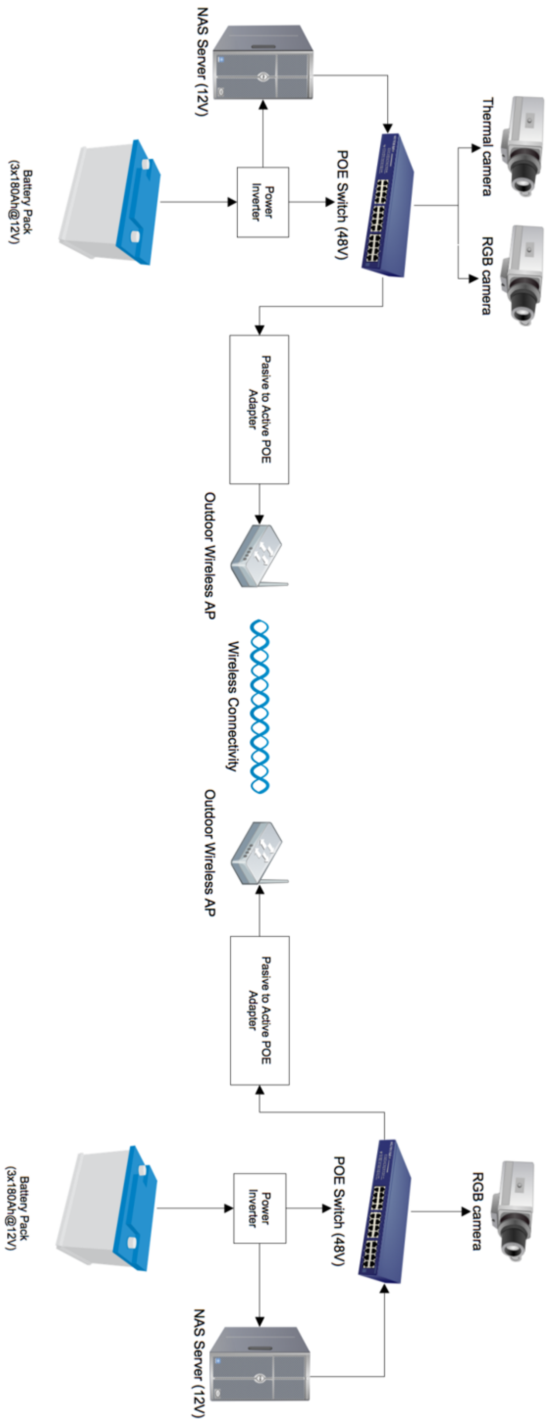


Fig. A.3: Overview of the developed video recording system.

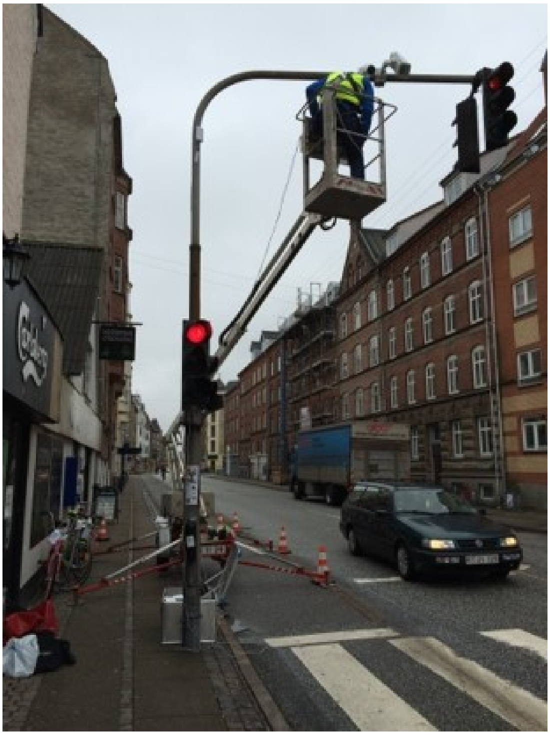


Fig. A.4: Cameras are installed in a traffic light pole. The cameras are connected NAS server inside the Eurobox located on the pavement next to the pole.

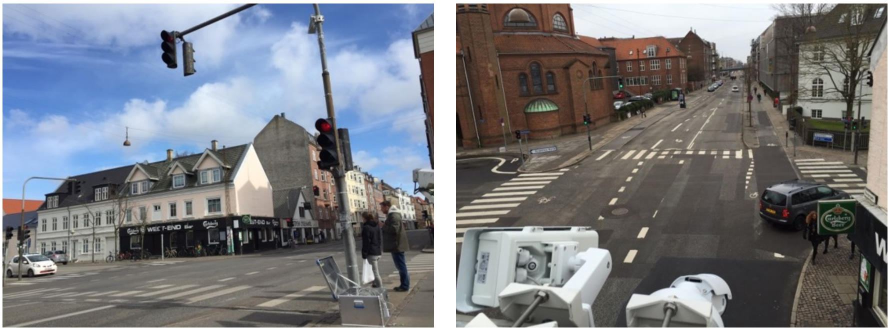


Fig. A.5: .

of the offset. The time is drifting quite a lot during these recordings, there have been observed a 4 seconds offset, which is quite a lot – and there might be worse examples in the large dataset that I have not examined. The NTP should under ordinary conditions adjust the clock based on offset in small steps so that the time is continuous, and hence "not broken". But the offset can be so large, that it bypass these small continuous adjustment steps and simply define a new time on the client equal to the reference clock on the server.

5 CONCLUSION

We have successfully developed a video recording setup that has been used for capturing video data within the InDeV Project. The video recording system has been deployed a large number of times and has thus been used to capture several months of traffic video data across Europe. The video recording system is however experiencing some problems maintaining a proper time synchronization at times. This is often and mostly caused by the wifi connection not being stable.

REFERENCES

[1] A. Laureshyn, A. Varhelyi, and Å. Svensson, "Project plan," in *InDeV: In-Depth understanding of accident causation for Vulnerable road users*, A. Laureshyn, Ed., 2015. [Online]. Available: https://www.indev-project.eu/InDeV/EN/Documents/pdf/project-plan.pdf?__blob=publicationFile&v=5

[2] C. H. Bahnsen, T. K. O. Madsen, M. B. Jensen, H. S. Lahrmann, and T. B. Moeslund, "The ruba watchdog video analysis tool," 2018.

[3] M. B. Jensen, M. Ahrnbom, M. Kruithof, K. Åström, M. Nilsson, H. Ardö, A. Laureshyn, C. Johnsson, and T. B. Moeslund, "A framework for automated traffic safety analysis from video using modern computer vision," *Transportation Research Board (TRB) 98th Annual Meeting*, pp. 201–213, 2019, in press.

Paper B

Collecting Traffic Video Data using Portable Poles: Survey, Proposal, and Analysis

Morten B. Jensen, Chris H. Bahnsen, Harry S. Lahrmann, Tanja K. O. Madsen, and Thomas B. Moeslund

The paper has been published in the
Journal of Transportation Technologies Vol. 8 No. 4, pp. 376–400, 2018.

The layout has been revised.

ABSTRACT

Several initiatives have been launched to help prevention of traffic accidents and near-accidents across the European Union. To aid the overall goal of reducing deaths and injuries related to traffic, one must understand the causation of the traffic accidents in order to prevent them. Rather than deploying a person to physically monitor a location, the task is eased by camera equipment installed in existing infrastructure, e.g. poles, and buildings, etc. In rural areas there is however a very limited infrastructure available which complicates the data acquisition. But even if there is infrastructure available in either the rural area or the urban area, this might not serve as an ideal position to capture video data from. In this work, we survey and provide an overview of available and relevant portable poles setups with respect to capturing data in both urban areas and rural areas. The conclusion of the survey shows a lack of a mobile, lightweight, compact, and easy deployable portable pole. We therefore design and develop a new portable pole meeting these requirements. The new proposed portable pole can be deployed by 2 persons in 2 hours in both rural areas as well as urban areas due to its compactness. The deployment and usage of the new portable pole is a complimentary tool, which may improve the camera capturing angle in case existing infrastructure is insufficient. This ultimately improves the traffic monitoring opportunities. Further, the survey of selected portable poles provides an excellent overview and can aid multiple applications within road traffic.

1 INTRODUCTION

Preventing traffic accidents and near-accidents remains a major and interesting challenge to address for academic partners as well as public organizations. In 2017 alone, the European Union (EU) reported that 25,000 people lost their lives and 135,000 people were injured on the roads across the EU [1]. In 2009 the EU estimated that the deaths and injuries across Europe costed the society approximately 130 billion Euro [2]. As a result, the EU set out a 2010-2020 goal with an overall objective of halving road deaths across Europe. To achieve this, several initiatives have been started covering increased enforcement of road rules, improved education and training of road users, safer road infrastructure, promote the use of modern technology to increase road safety (ITS), and protection of vulnerable road users (VRU). All of which are important to analysis and address to meet the overall objective in 2020.

Understanding accidents causes in the traffic requires a lot of data, which can be collected with different purposes. Naturalistic Driving Study (NDS) such as the "100-Car Naturalistic Driving Study" [3] and the "SHRP2 Naturalistic Driving Study" [4], collects all sorts of data from within the participating vehicles such as GPS, accelerometer and similar vehicle network data, but the vehicles are also equipped with multiple different sensors, e.g. RGB cameras, thermal cameras, stereo cameras [5] or radars. Though these studies generate a lot of interesting data, a major drawback of this approach is the large investments needed to reach a large participant pool and then afterwards installing expensive equipment inside the car whilst keeping the car naturalistic.

A less expensive approach of capturing data that helps understanding accidents causes is simply to monitor and observe a critical location, e.g. traffic intersection. This manually task is however quite error-prone as the assigned person must be aware of everything happening in area of interest whilst continuously documenting the observations over a longer period of time. So rather than deploying a person to physically monitor a point of interest, the task is eased by mounting a camera-based system in existing infrastructure, e.g. poles, and buildings, etc. The captured video data can then be post-processed and analyzed with the purpose of understanding the scene and ultimately making adjustments that ideally prevents accidents and near-accidents. The main challenge of the camera-based system is that often there is no or very limited existing infrastructure available at the scene, thus directly impacting the quality of the analysis. This has spawned the use and interest in portable setups that can be moved around, which allows for a more optimal data collection in both urban areas but in particularly also in rural areas where there is often no proper infrastructure to mount cameras in.

In this paper, we make an analysis of relevant portable setups, where we discuss the pros and cons of different portable types and solutions, thorough overview of available setups. The result of the overview shows a lack of a mobile, lightweight, and easy deployable portable pole, thus we design and develop a new portable pole meeting these requirements.

The contributions of this paper are thus twofold:

1. Providing a thorough analysis and overview of available portable camera-based capturing setups.
2. Design and development of a new mobile, lightweight, and easy deployable portable pole to ease camera-based data collection.

The paper is organized as follows: Section 2 describes the minimum requirements for the portable pole as well as the general definitions used. All of the requirements and definitions are then used examining various solutions ultimately providing an overview of available portable pole solutions in Section 3. In Section 4, the design and development of the new portable pole is presented. Usage and applications of new portable pole is presented in Section 5. In Section 6 we perform a discussion of our work. Finally, we present our conclusions in Section 7.

2 PORTABLE POLE ANALYSIS

Portable poles can serve multiple purposes and can be used for various applications. As briefly mentioned and introduced in Section 1, this survey will only consider portable pole solutions that could be relevant as a camera-based recording platform in the field of traffic surveillance and monitoring.

2.1 Minimum Setup Requirements

The relevant portable pole solutions are derived based on 4 minimum requirements that are considered essential for a portable pole to function as a proper camera-based recording platform, which can be utilized in both urban and rural traffic environments.

Recording Time

The video recordings are the basis for the entire analysis, so besides having a great view-angle provided by either the infrastructure or a portable pole, the video recordings must contain a sufficient amount of accidents or near-accidents in order to make some concluding remarks of a given location. In [6], the frequency of traffic accidents is described as a pyramid, where the pyramid base contains normal traffic encounters that are non-critical and rather safe, but very frequent. The pyramid apex contains the fatal and very severe events, e.g. fatal injuries, these are however occurring more infrequent compared to accidents in the lower part of the pyramid. Previous studies from Scandinavia show that at a particular site, the number of near-accidents tends to be as low as 1-2 per day [7] [8] [9]. So in order to get video recordings containing some infrequent events, the portable pole and camera-based setup must robust and stable enough to record continuously throughout a longer period of time. In this analysis we consider a period of 3 weeks to be the minimum requirement.

Capturing Height

A major issue to take into account when installing camera equipment at a point of interest is occlusion. Occlusion is in this case defined as when two objects are overlapping each other from the view-angle of the camera equipment, which makes the objects completely or partly occluded. In Figure B.1 an example of this is shown, where the red car is clearly not visible from the specific camera-view mounted in existing infrastructure.

To reach the most accurate conclusion in a traffic analysis, the data needs to be as accurate as possible, thus we want as little occlusion as possible in the data collection. There are multiple ways of reducing occlusion, e.g. having multiple cameras from different view-angles or simply just by increasing

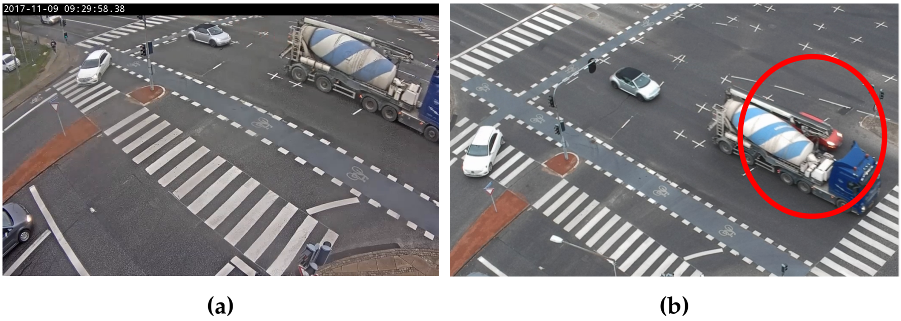


Fig. B.1: Objects can overlap each other in the camera-view as seen in (a) where the the large cement truck clearly occludes the lane behind it. (b) clearly shows that a red car is in fact driving side-by-side of the cement truck.

the capturing height similar to the Figure B.1b. In this analysis, we define a minimum capturing height of 7 meters for the portable pole, which is 3 meters higher than the maximum height limit for vehicles in most countries in Europe [10] [11].

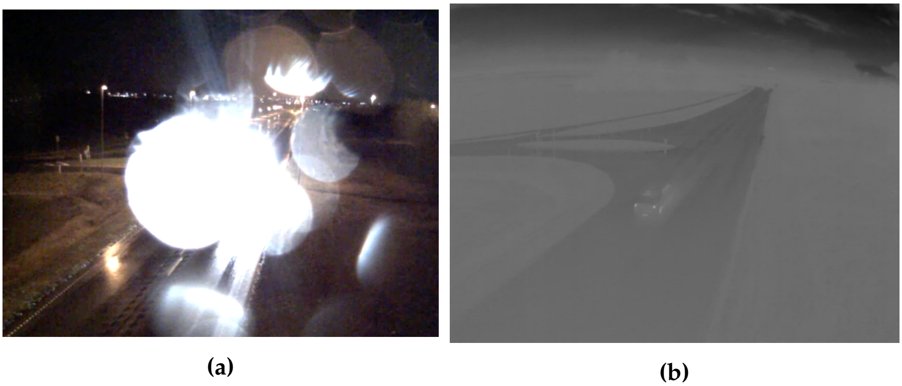
Ground Area Occupation

To make sure, that the data collection is done in an as naturalistic and unobtrusive environment as possible, we need to make sure that the base does not cause any major impact on the behavior of the drivers on the road or the pedestrians on the sidewalk. Naturally, placing a new "intruder" in an existing environment may attract some attention and thus result in changed driver behavior, but the point of this demand is to keep it at a minimum by defining the maximum ground area occupation of the portable base to be 1.5 meters in the width. This should enable deployment of the portable pole in rural areas and in most urban environments as it can be deployed on the sidewalk whilst pedestrian should be able to easily walk around it. The maximum ground area occupation is only defined for the width, as this is the strictest one in terms of occupying the sidewalk. The length is less critical as people are still able to use the sidewalk, however it should preferably be under 2.5 meters.

Payload Weight

The portable pole setup must be able to handle the payload weight from the capturing devices mounted in the top. In this analysis, we suggest using both a RGB camera and thermal camera as capturing devices. Using multi-modal visual cues provides a solid data foundation for a later accident causation analysis as accidents and near-accidents do not solely happen in daylight

[12]. Doing periods with a limited amount of light and challenging weather conditions, e.g. night, winter, rain. Thermal cameras are quite useful as illustrated in Figure B.2, where both modalities are seen showing the same scene.



(a) (b)

Fig. B.2: Data collection at 02:00 in the night using two modalities: (a) RGB camera (b) Thermal camera.

The RGB camera is having a hard time coping with the headlights from the car and the low-light in the reminder of the scene. Furthermore, the RGB camera seen in Figure B.2a is challenged by the weather conditions, i.e. rain. The thermal camera on the other hand do not rely on light to produce its output but infrared radiation, which clearly produce a more accessible output as seen in Figure B.2b, where the car is clearly visible. The pole must therefore be able to handle a setup with two capturing devices. The capturing devices in this analysis are seen in Table B.1, which defines a minimum payload weight requirement of 5.7 kg.

Table B.1: Derivation of the minimum payload weight requirement using AXIS RGB camera and thermal camera.

Type	Manufacturer	Model	Weight [kg]
RGB	Axis	Q1615-E	3.5
Thermal	Axis	Q1932-E	2.2

Below are the requirements for a portable pole listed, if nothing else is stated, these are minimum requirements.

1. Solution must be able to record continuously in 3 weeks.

2. Capturing height: 7 m.

3. Maximum ground area occupation(Width): 1.5 meters.

4. Payload weight: 5.7 kg.

2.2 Portable Pole Types

In this analysis, we have divided portable poles into 4 different types, which will also form the structure for the reminder of the portable pole analysis and overview, namely: 1) Lightweight and compact portable pole with low payload; 2) compact portable pole with high payload; 3) trailer portable pole with high payload; and 4) heavyweight portable pole with high payload.

The payload is the capacity which the portable pole is able to lift in the top during operation. The stability in the top of the pole, hence the recording usage quality, is dependent on the payload. Common for all of the portable pole types are that they all must comply with the minimum requirements defined in Section 2.1.

Type-1 *Lightweight and compact portable pole with low payload:* The main goal of this type is that they are very easily moved and transported between locations. The efforts needed for setting up this type of portable pole is very low. The setup and transportation of this type of portable pole is a one-person job, requiring it to be lightweight and compact. The stability and payload scales accordingly, resulting in a low payload to keep the pole stable in the top.

Type-2 *Compact portable pole with high payload:* Rather than being able to transport the portable pole by yourselves, this type consider more heavyweight equipped that can be assembled on-location by one or two persons. The equipment will remain compact while dissembled such it can be easily transported from location to location by use of a van or pick-up truck. When assembled the equip-ment is more robust compared to type-1, but at the cost of easy mobility.

Type-3 *Trailer portable pole with high payload:* This type utilizes a trailer or small wagon which can be attached to a vehicle's hitch ball. All the equipment is installed upon this trailer, such that one or two persons can drive to a location and set up the portable pole without too much assembling and more lenient requirements for the level of the ground base. This provides a rather stable portable pole with some degree of mobility.

Type-4 *Heavyweight portable pole with high payload:* By using a large platform of e.g. concrete, all the equipment can be installed on this providing a robust platform for the portable pole. However, this require a large truck with a crane for transportation, but provides a good pre-assembled portable pole.

This division will form the structure for the portable pole overview section when surveying the corresponding available portable poles.

3 OVERVIEW OF RELEVANT PORTABLE POLES

The overview is divided into 6 parts. The first part introduces a general base framework that complies with the battery and storage requirements and is applicable for most of the portable poles presented. This is followed by 4 parts, one for each of the 4 portable pole types presented in Section 2.2. The final part presents an overview that summarizes all of the presented portable poles.

3.1 Base framework

Regardless of the portable pole choice, the data recording capacity, the power supply and underlying video acquisition framework must fulfill the minimum requirements. Using aforementioned minimum requirements, we will in this subsection define a common framework that can be used together with the portable poles.

Video acquisition

The Axis cameras defined in Table B.1 are capable of operating by Power over Ethernet (PoE) which means it is only necessary to supply one cable per camera in the mast. The cameras are by the use of a network switch connected to a Synology DS215j Network Allocated Storage (NAS) server, where the acquired video data must be properly stored. The storage capacity required is heuristically derived to be no less than 6 TB in order to keep 3-weeks of data using H.264 compression.

Power supply and enclosure

The video acquisition hardware presented above must be powered throughout the 3-week acquisition period. The power supply and some of the video acquisition hardware must also be placed in an enclosure which is resistant to tampering.

The video acquisition hardware consumes approximately 30 watts in operation, which make a self-contained setup unfeasible due to 3-weeks video acquisition requirement. Instead we use 3 heavy-duty 12 volt 180 Ah batteries, which provides the setup with an approximately replacement cycle of 4-6 days depending on the overhead and wear out of the batteries. The entire

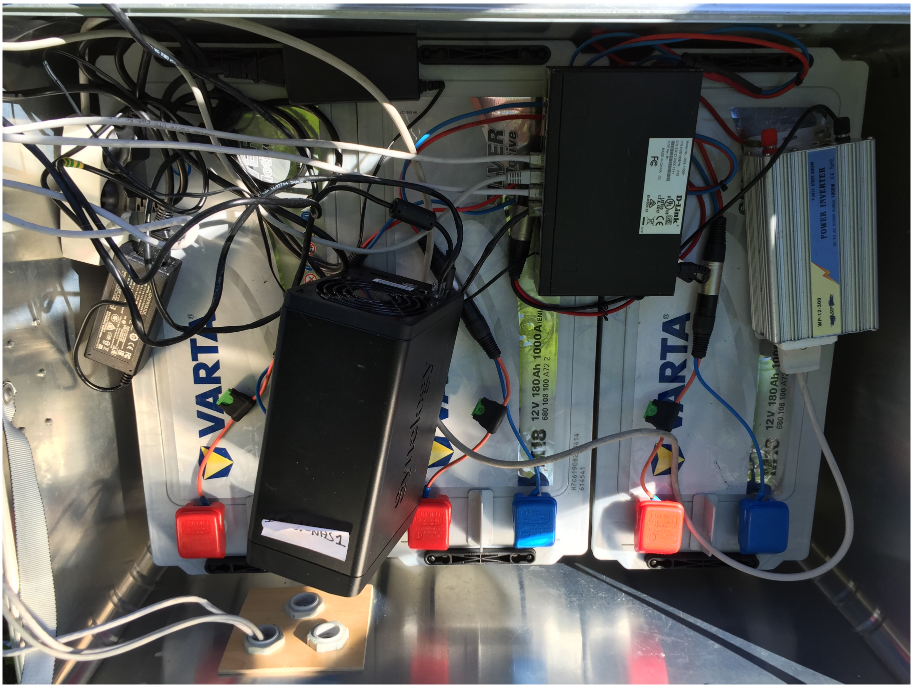


Fig. B.3: The Eurobox 40705 containing 3 batteries, a 230V power inverter, a PoE switch, and a NAS server.

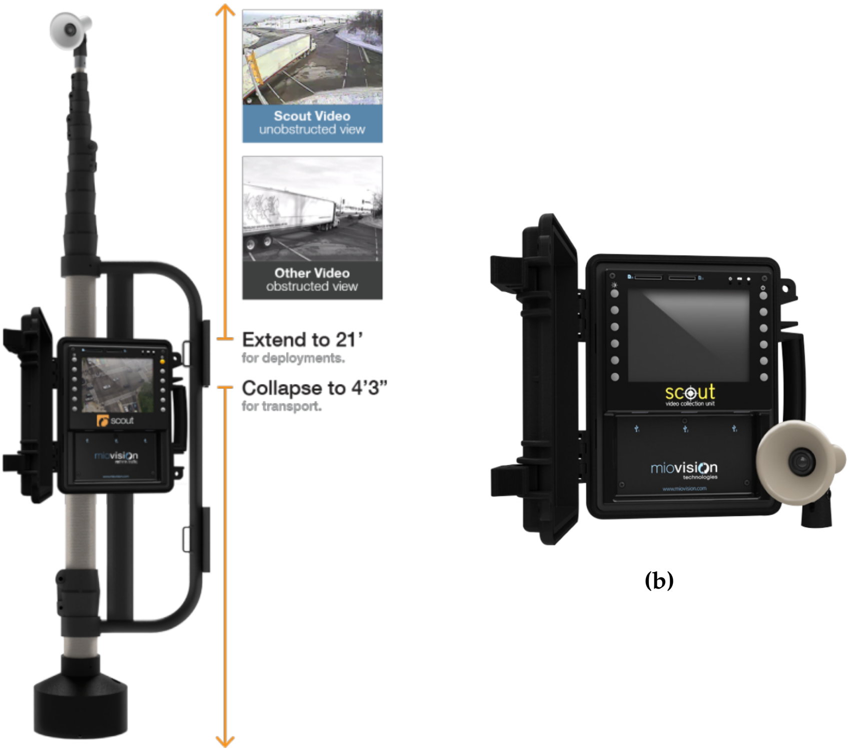
system is finally installed in an IP65-certified Eurobox 40705, which can be seen in Figure B.3.

3.2 Type-1: Lightweight and compact portable pole with low payload

The first type of poles, is as introduced in Section 2.2, the most compact and lightweight ones, and should ideally be deployable for a single person.

Miovision Scout

Scout is a portable and expanded pole developed by Miovision, and is, according to their own documentation, "designed specifically with the users in mind" [13]. This has resulted in a portable pole with a weight of only 19.1 kg and a set up time of 10 minutes. The Miovision Scout do not meet the requirements for this analysis, defined in Section 2.1, as it is not configurable for the two cameras defined in Table B.1. It is however still included as it is a very popular solution for traffic monitoring, and might be usable in pilot tests or as a second view-angle.



(a)

(b)

Fig. B.4: The lightweight and compact, but non-configurable, portable pole from Miovision. (a) Miovision Scout with the extendable pole [13] (b) Miovision Scout Video Collection Unit. [13]

The Miovision Scout has a battery life of 7 days when buying the additional power pack and can be set up on existing infrastructure using an included pole mount. The simplicity of the product can easily be deducted by examining Figure B.4. In case deployment is needed in places without street poles, a separately sold Scout Tripod can be used. The Scout Tripod weights 14 kg, but can reach 68 kg with additional security weights. The Miovision Scout is equipped with a wide lux camera with 120°horizontal view capturing with a resolution of 720x480 pixels @ 30 FPS. As mentioned in the introduction, this camera setup is not configurable. The operational height can be adjusted to be between 1.32-6.4 meters, which do meet the requirements either. In Table B.2 an overview of the required equipment is seen.

The Miovision Scout can be mounted to existing infrastructure, such as a pole, defining some requirements to how poles or similar objects are located at an intersection. Otherwise the Scout Tripod can be used to deploy the

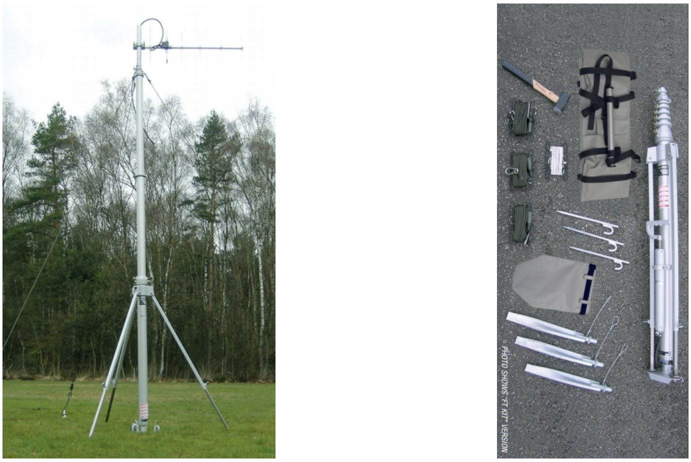
Table B.2: Required equipment for the Miovision Scout.

Product	Weight
Scout Video Collection Unit	10.89 kg
Scout Pole Mount	8.16 kg
Scout Power pack	14.0 kg
Scout Tripod	14.0 kg

Scout. For both of the solutions no major equipment is needed, and one person should be able to set this up in an hour.

Custom lightweight portable pole

This portable pole is a proposal on how a lightweight portable pole could be manufactured. The portable pole must meet the requirements defined in Section 2.1, while being a lightweight solution easily transported around.



(a)

(b)

Fig. B.5: Parts for a custom lightweight portable pole (a) Clark Mast FT mast. (b) Clark Mast FT carrying bag. [14]

The portable pole utilizes the Clark Masts SFT9-6 mast, which can be extended to 8.8 meters using a hand pump, and remains at 2.05 meters in retracted mode. An image of a FT series mast from Clark masts is seen in Figure B.5a. On top of this there must be created a rig which cameras can be mounted in. The mast comes with a carrying bag, seen in Figure B.5b for easier transportation. In addition to the bag, equipment such as spikes and radius lines are also included. This solution must also utilize the base framework presented in Section 3.1. In addition to the base framework, Table

B.3 summarizes the additional required equipment to manufacture this type of pole.

Table B.3: Equipment needed for custom lightweight portable pole. An unknown weight is marked with a "-".

Type	Model	Weight
Telescopic mast	Clark Masts FT series, SFT9-6/HP 10 kg headload, 8.80 m extended height, 2.05 m retracted height, w. tripod	-
Carrying bag	Clark carrying bag, SFT9-6/HP Bag	-

A van must be used to transport the equipment from location to location as the mast is 2.05 meters long, but setting up the equipment should be doable for one person. Using the radius line to make a guying system is however not really feasible in urban places, requiring the wind speed to be low for the setup to remain usable.

Discussion

The Miovision Scout do not meet the configurable requirements defined in Section 2.1 and can therefore not be used in the final setup. It might, however, be a useful solution for some minor pilots tests or be used a second view-angle at a complex environment. The custom made portable pole is not as lightweight as the Miovision Scout as one needs to bring more equipment to meet the requirements of capturing data continuously in 3 weeks. The custom made portable pole can be configured to have 2 cameras installed, but it is however considered necessary to utilize a guying system in order to stabilize the portable pole sufficiently, even in low wind conditions, such the video recordings are stable and usable for a traffic analysis.

3.3 Type-2: Compact portable pole with high payload

We divide possible solutions for systems using a compact portable pole with high payload into three proposals based on the estimated total weight of the system: lightweight, middleweight, and heavyweight. All of them utilize the base framework presented in Section 3.1.

Lightweight: Mast with tripod

The lightweight portable solution consists of a telescopic, 5-section mast with a corresponding tripod. The extended mast is usually secured by a guying system to assure stability under heavy payload and wind speeds. However, as guying is not applicable in urban areas, we include a tripod to ensure stability. The tripod furthermore ensures independence of existing infrastructure and comes in a variety of sizes for different mast heights. An image of such a pole is seen in Figure B.6.

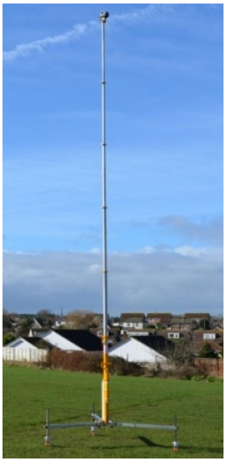


Fig. B.6: Clark QT Mast on tripod [15].

We choose the largest mobile tripod available to provide stability and accommodate the requirements even under moderate wind speeds and payloads. The base diameter of the tripod is 2 m, which have a recommend maximum mast height of 10m. When the mast is not guyed, the maximum wind speed is 13.8 m/s for stable operation. A wind speed of 13.8 m/s translates to 'Strong Breeze' on the Beaufort scale.

The necessary equipments for the lightweight mast with tripod are listed in Table B.4.

Table B.4: Required equipment for compact, lightweight portable pole with high payload. Maximum wind speed 13.8 m/s. An unknown weight is marked with a "-".

Product	**Model**	**Weight**
Telescopic mast	Clark QT Series, SQT9-5, 5 section mast 18 kg payload, 9.00 m extended height, 2.25 m retracted height	-
Tripod	Clark MK VI, MK6 2000MM	18.0 kg
Tripod adapter	Clark	-

The transportation of the equipment requires a medium-to-large sized car or van to accommodate the length of the retracted mast and the total weight of the equipment. The telescopic mast is extended by an integrated hand pump, and the extended section is subsequently locked manually by using the provided screws. The ground area required for the base is 0.5 m larger than specified in the setup requirements. The extra space is however necessary for the stability of the portable pole.

Middleweight: Mast with tripod

The lightweight setup, described in Section 3.3, is used as a point of departure for the middleweight portable setup where the telescopic mast and tripod remain key components. The Clark QT mast from the lightweight setup is replaced by the heavier and sturdier NT series and features only 4 sections compared to the 5 section QT mast. The heavier mast calls for a heavier and larger tripod which is found in the Clark MK IV Tripod. The tripod weighs 27 kg and features a base diameter of 2.6 m. As with the light-weight mast with tripod, the un-guyed mast is stable up to wind speeds of 13.8 m/s. The equipment of the middleweight mast with tripod is listed in Table B.5.

Table B.5: Required equipment for compact, middleweight portable pole with high payload. Maximum wind speed 13.8 m/s. An unknown weight is marked with a "-".

Product	Model	Weight
Telescopic mast	Clark NT Series, NT 90-4, 4-section mast, 15 kg payload, 9.00 m extended height, 2.82 m retracted height	41.0 kg
Tripod	Clark MK IV	27.0 kg
Tripod Adapter	Clark	-

Due to the larger retracted height of the telescopic mast (2.82 m) it might be impossible to fit inside an ordinary car, and thus a larger van is recommended. The telescopic mast is extended by the use of a hand pump and the sections are secured by screws similarly to the lightweight setup. The ground area required is even larger than for the lightweight scenario; however, this is needed in order to provide stability for the heavier mast.

Heavyweight: Flyintower

The heavyweight compact portable pole solution uses a Flyintower, or sound tower, as the camera mast. The Flyintower is a well-known object at large concerts or festivals where it is used for the lifting of loudspeakers as depicted in Figure B.7. The V-shaped basement, the metal grid, and the heavy weight of the construction improve the sturdiness and stability of the setup.

We choose the smallest possible Flyintower from Litec to minimize the ground occupation area required for the basement of the tower. The ex-

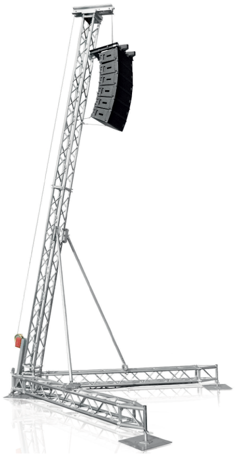


Fig. B.7: Litec 7.5-500 Flyintower [16].

tended height of the tower is 7.75 m and due to the V-shaped basement, the footprint is 4.1 × 3.6 m. The maximum lifting load capacity of the tower is 500 kg which requires additional ballast at the base for stability. For the much lighter loads required in this setup, the required ballast weight is reduced. Due to the studier nature of the setup, the maximum wind speed is increased compared to the lightweight and middleweight setups. A list of the equipment is found in Table B.6.

Table B.6: Required equipment for compact, heavyweight portable pole with high payload. Maximum wind speed 70 km/h. An unknown weight is marked with a "-".

Product	**Model**	**Weight**
Flyintower	Litec 7.5-500, 500 kg max load capacity, 7.75 m extended height	160 kg
Ballast	Required ballast for Flyintower	-

The Flyintower is considerably heavier than the mast-based solutions listed above. However, the tower might be taken apart and assembled on-site which greatly reduces the space needed for storage and transportation. We therefore estimate that a larger van is needed for the transportation, just as in the middleweight scenario. Compared to the lightweight and middleweight scenarios, the Flyintower requires a larger, planar surface for the base to stand. This might exclude the deployment in tight urban spaces where such space is not available.

Discussion

For both the light portable poles and middleweight portable poles, issues arise when dealing with higher wind speeds as the equipment is mounted in the top making the setup unstable. To cope with this, a guying system can be installed to stabilize the mast, this is however not feasible in urban places. For most scenarios in urban environments, both portable pole setups are considered usable in terms of wind speeds. The heavyweight solution is therefore a better overall option due to increased stability, but significantly comprising the compactness and weight compared to the lightweight and middleweight solutions. Generally, all of the solutions can possibly be disassembled and be somehow compact and then be used in rural areas where there are more open space, it is however not ideal that none of the proposals meet the maximum ground occupation area requirement. Deploying any of the introduced solutions in this section in an urban environment will most likely be considered unnaturalistic and obtrusive.

3.4 Type-3: Trailer portable pole with high payload

The third type of portable poles differs from the both type-1 and type-2 in the sense that the equipment used comes in a more wrapped up and easy-deployable way. As mentioned in Section 2.2, type-3 relies on equipment installed either in a trailer or in a small wagon resulting in less assembling on-site.

UTRaCar

The Urban Traffic Research CAR is developed for the national aeronautics and space research center of the Federal Republic of Germany (DLR) [17] and is equipped with a large set of sensors and systems to be used for traffic surveillance and data acquisition in the field. The car is seen in Figure B.8a in transportation mode and in Figure B.8b where the left image show an image of the car in operation [18]. The UTRaCar does not meet the requirement of the maximum ground occupation area, but is included as it provides some interesting solution ideas.

The car is equipped with multiple sensors as seen from the images in Figure B.8b. For this analysis, the telescopic mast seen in the left image is the most interesting one. A telescopic mast is mounted in the back of the car, and can extend to 13 meters. In the top of the telescopic mast various sensors can be installed, as seen in the upper right image in Figure B.8b. According to [14], the power supply unit in the car is self-sufficient. It is unclear what this covers, but from the lower image in Figure B.8b, it is clear that a lot of equipment can be installed in the of the car. In Table B.7 an estimate of the equipment needed for a minimum requirement solution are seen.

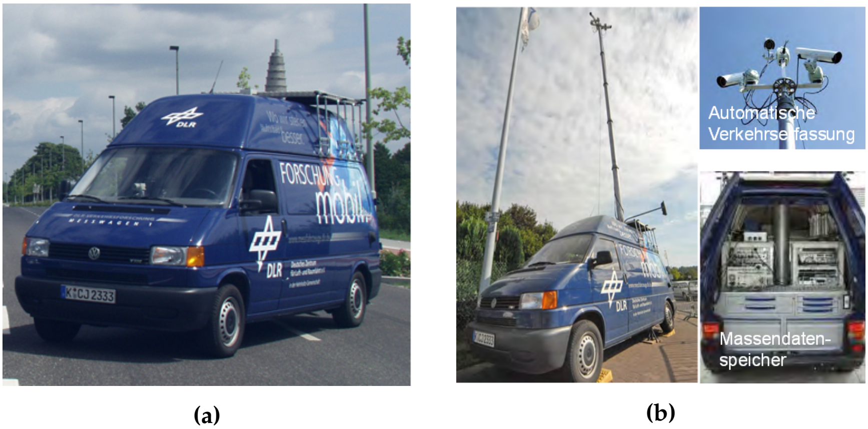


Fig. B.8: (a) The DLR UTRaCar with retracted telescopic mast. (b) The DLR UTRaCar with extracted telescopic mast. [18]

Table B.7: Estimated equipment needed for a minimum requirement version of the UTRaCar. An unknown weight is marked with a "-".

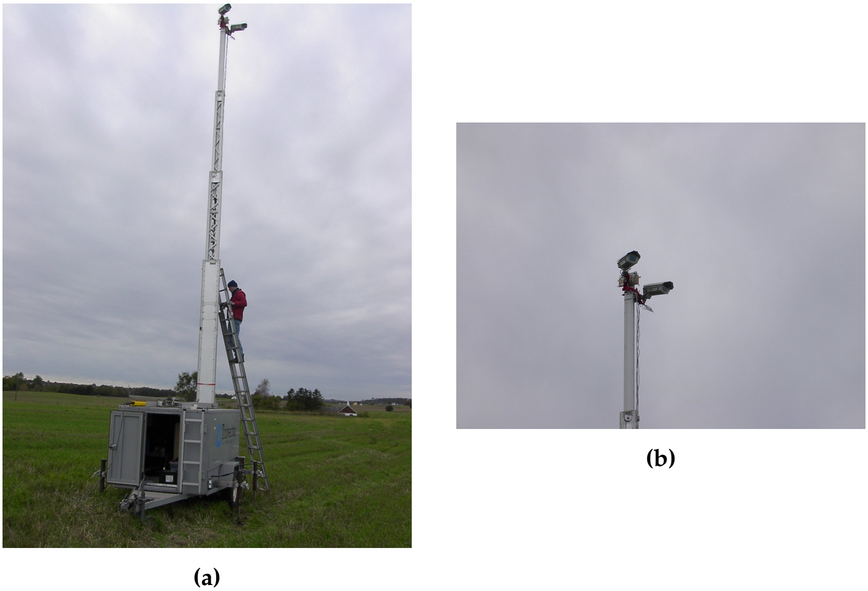
Type	Model	Weight
Van	VW Crafter 35 with medium wheelbase and high roof	-
Telescopic mast	Clark WT Series, WT100-4, 4 Section mast, 140 kg headload, 10.0 m extended height, 3.32 m retracted	-

The size of the car can be a challenge at a lot of intersections, so there must be some open areas around the intersection for deploying this system. But if the area suffices, a solution like this allows a rather fast deployment without any external actors.

Trivector Mobile Mast

The Swedish based company Trivector has developed the TMV1, which is a mobile mast installed in a trailer with the scope of capturing traffic situations. When extracted the height can reach up to 15 meters. In Figure B.9a an image of the setup is shown, and in Figure B.9b it is visible that the setup utilizes two cameras in operation meeting the requirements for this analysis.

The setup consists of a trailer equipped with a custom made telescopic mast. Inside the trailer all the equipment can be stored, and given from the image seen in Figure B.9a, it is clear that box is rather large, providing good possibilities to put all equipment inside. There exists no technical data sheet available to the public, hence it is hard to estimate the equipment used to cre-


(a)

(b)

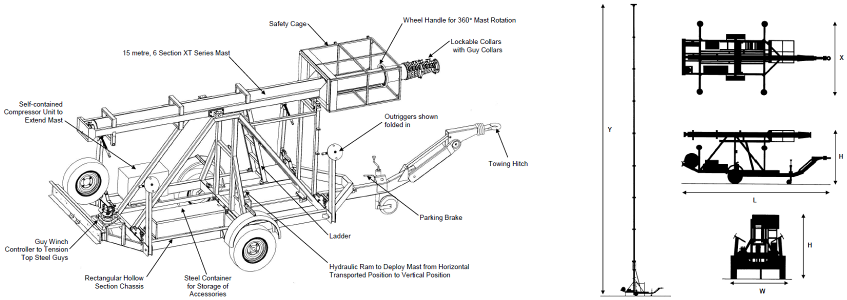
Fig. B.9: (a) The Trivector Mobile mast setup in operation. (b) The Trivector Mobile mast with two installed cameras. *Images provided by Aliaksei Laureshyn, Lund University.*

ate the Trivector mobile mast. From examining the figures the minimum requirements are a cargo trailer and a telescopic mast. As for the UTRaCar, the setup occupies a rather large area on the ground, making it difficult to place in some urban areas. The installation complexity is low as it all equipment are inside the trailer, so the deployment is straightforward with a minimum of external actors. Finally, a car is needed to tow this setup from point A to point B.

Custom made trailer

With inspiration of the previously solutions in type-3, we look into to assembling a trailer portable pole. The main idea is to utilize a trailer solution with a pole mounted on it. In Figure B.10a and Figure B.10b the main component in the setup is seen. It consists of an already existing product which needs to be customized to accommodate the minimum requirements. Though there are some boxes and containers mounted in the original Clark Mast 804-15-6, additional room is considered necessitated to meet the capacity requirements. The 6 section XT Series mast mounted on the trailer has an extracted height of 15 meters. [19]

A vehicle is needed to tow the trailer from location to location. A regular van is considered to be sufficient to tow the trailer and the remaining equip-



(a) **(b)**

Fig. B.10: Specifications and overview of Clark Mast 804-15-6. [19]

ment The length of the trailer is 6.3 meters, making it quite large and difficult to deploy in tight urban spaces.

Discussion

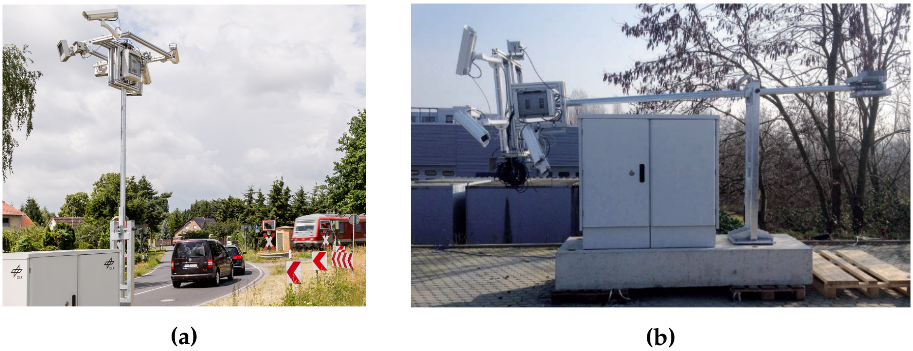
For all of the trailer solutions the main advantages are the easy and rather fast deployment as a very limited amount of external actors are needed. Other advantages of the type-3 solutions are a "all-in-one" solution in the sense that room for batteries, HDD and other equipment is included in the setup. The disadvantages are that they can rather fast become quite expensive, and they do not scale very well in the sense of transportation from point A to point B might require multiple cars or a large truck. The weight and size of the solutions also occupies a large ground area which challenges one of the requirements defined for this analysis. Furthermore, a regular driver's license might not be sufficient for all of the solutions. The UTRaCar solution is considered to become quite expensive to build, so the best option in type-3 is to use a solution similar to the Trivector mobile mast or the custom made trailer, even though it is also expected to become expensive.

3.5 Type-4: Heavyweight portable pole with high payload

The last type of portable poles takes it starting point in a large platform of e.g. concrete, where all equipment can be installed upon. This complicates the transportation phase but should have advantages in operation compared to the previous types.

DLR Platform

The National Aeronautics and Space Research Centre of the Federal Republic of Germany (DLR) have used a portable platform for data capturing. The development of the technical aspects of the portable platform was carried out by Jenoptik. One of the usages of it has been to monitor railroads crossings as seen in Figure B.11a. The camera equipment used in the DLR portable pole setup consist of 4 cameras, 2 IR-flashes, 2 radars, and an aluminium frame, totaling a payload weight of 25.4 kg. In operation mode, the camera fixed to an operational height of approximately 4-5 meters. [20] As seen from Figure B.11 it is clear that the entire portable pole consists of a cabinet and a port that is split into two pieces, and is mounted onto a large concrete block. In operation mode, the pole is angled in vertical position, opposite to the horizontal transportation angle seen in Figure B.11b. The equipment needed for creating a portable pole for meeting the minimum requirements are seen in Table B.8. The mast could however be changed to a telescopic mast.



(a) **(b)**

Fig. B.11: (a) DLR Setup in operation mode. (b) DLR Setup in transportation mode. [20]

Table B.8: Required equipment for heavyweight portable pole with high payload mounted on a concrete block. An unknown weight or model is marked with a “-“.

Type	Model	Weight
Custom mast	Mast divided into two parts: Transportation and operation mode	-
Cement block	-	-
Vandalism-proofed cabinet	-	-

According to an interview with Kay Gimm and Sascha Knake-Langhorst from DLR, it can require up to a whole day to setup and calibrate the sensors for the specific application. As the current system requires power supply access from the current infrastructure. Due to that concrete block, the setup is quite heavy requiring a truck and crane to move it around.

Discussion

Only one solution is presented for the type-4, which is the DLR setup. This setup provides a good and solid platform for data capturing. Installed on the concrete block is all the equipment needed making it a "all-in-one" solution. However, the setup is heavy meaning a truck and crane is needed for transportation and deployment.

3.6 Overview

Creating a setup that is lightweight, robust, and as mobile as possible is a hard problem to satisfy. It might become easier to record traffic data at certain intersections if using a small and lightweight setup. One can, however, not be certain of the quality of the recordings as lightweight usually correlates with instability during varying weather conditions; especially when considering that the setup has a relatively heavy camera rig mounted in the top. All of the surveyed options are seen in Table B.9.

The main parameter to satisfy is considered to be the recording quality, as the quality of the data is essential for performing a good traffic analysis. Taking this into account, the proposed solutions from both type-1 and type-2 are not good options as they require guying systems in order to reach stability for prolonged periods of time. Guying systems are not ideal in urban environments, and the lightweight and compact pole solutions examined in this analysis does therefore not pose an ideal fit for the requirements.

For both type-3 and type-4, the solutions presented will provide some more stable recording platforms however they are considered quite expensive to produce, and does therefore not scale very well. Furthermore, the solutions of type-3 are in most cases wider than the specified maximum of 1.5 meters, hampering the deployment on the sidewalk without interrupting the pedestrians. The type-3 solutions are, however, more mobile compared to the type-4 solution, but in both cases a regular driver’s license might not be sufficient. Additionally, the trailer option does not scale well as multiple trailers requires multiple towing vehicles.

This leads to the conclusion that for capturing the most stable and useful data, the setup must comprise the lightweight and easy mobility requirements. For type-1 and type-2 solutions to work, various guying system must be installed on existing infrastructure to fixate the pole. If one involves the existing infrastructure, a better result would be reached if the capturing rig is mounted on the infrastructure rather than using a light-weight or compact portable pole with guying installation. The type-4 solution from DLR requires both a truck and a crane to deploy, which satisfies most of the requirements for this analysis, but remains, however, the less mobile solution in this analysis.

Table B.9: Overview of the analyzed portable poles. The poles are summarized and can easily be compared on the 7 different parameters.

Type	Type Name	Operational height [m]	Payload [kg]	Operational base dimensions [L x W x H [m]]	Transport dimensions [L x W x H [m]]	Weight [kg]	Configurable	Deployment equipment
T-1	Miovision Scout [13]	1.3 - 6.4	-	1.5 x 1.5 x 1.24	-	48	No	Car
	Custom lightweight portable pole	8.8	10	-	2.05 x 0.25 x 0.25	-	Yes	Van
T-2	Lightweight: Mast with tripod	9.00	18	2.0 x 2.0 x 9.0	2.25 x 0.4 x 0.4	-	Yes	Car
	Middleweight: Mast with tripod	9.00	15	2.6 x 2.6 x 9.0	2.8 x 0.4 x 0.4	68	Yes	Car
	Heavyweight: Flyintower	7.75	500	4.1 x 3.6 x 7.75	-	>160	Yes	Van
T-3	UTRaCar [17]	13	-	5.9 x 2.4 x 2.4	5.9 x 2.4 x 2.4	>2800	Yes	-
	Trivector Mobile mast	15	-	-	-	-	Yes	Car
	Custom made trailer	15	140	6.3 x 1.95 x 15	6.3 x 1.95 x 2.2	-	Yes	Car
T-4	DLR Platform	4-5	25.4	-	-	-	Yes	Truck, crane

4 DESIGN & DEVELOPMENT OF TRG-POLE

In this section, we will present a pole which is hybrid between a type-2 and type-4 portable pole solution designed specifically to contain the same advantages as the DLR solution while being mobile.

4.1 The designed pole

We present a portable pole design that accommodates the overall portable pole goal while being in operation mode. It is, however, desirable to keep the weight down during transportation. To reach this, we propose creating the pole as a hybrid between a type-2 and type-4, meaning that the pole is compact and has a reduced weight during transportation, but which in operation mode remains robust and stable. One of the main weight contributors in the DLR setup is the concrete base which the entire pole is installed on. Naturally, a proper frame is needed to keep the base stable, however, all additional weight needed should be configurable. In Figure B.12 the proposed base design of the portable pole is seen.

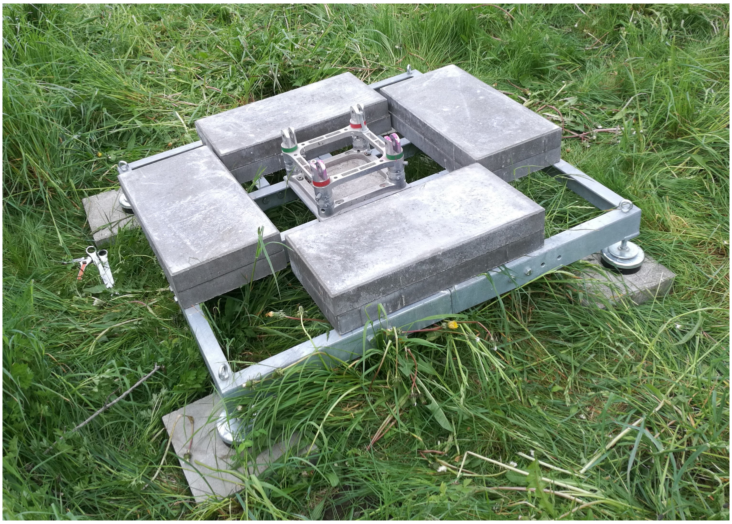


Fig. B.12: The ground base of the portable pole is equipped with tiles, adjustable feet, and a swivel bracket to ease the raising of the lattice mast.

The entire square platform consists of a steel frame containing 4 slots for mounting standard tiles in a vertical rack. The tiles can be acquired in most construction and hardware stores around the world, i.e. 30x60x6cm tiles with a weight of 25kg each. Depending of the required base weight, one of the tiles slot could be used for the equipment cabinet rather than placing it next to the base. Finally, the base platform has 4 adjustable feet for levelling its height on site in case the pavement is not well levelled.

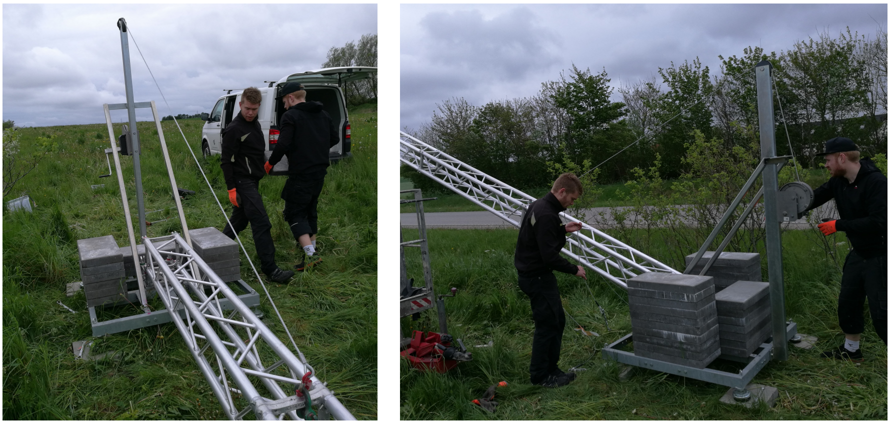


Fig. B.13: The portable pole can be raised using a swivel bracket installed in the middle of the base platform. The pole is raised using a steel wire connected to a manual winch system.

The swivel bracket installed in the middle of the base platform will be used for raising the lattice mast as seen in Figure B.13. The deployment of the portable pole is done by in-stalling tiles in 3 of tiles slots on the base platform leaving 1 slot open. The lattice mast is connected to the swivel bracket in the center of the base platform and put horizontally on the ground in the open tiles slot direction. Our portable pole consists of 5 lattice mast sections, which are 2 meters each providing a 10 meters long lattice mast. The lattice mast and base can be completely separated to ease transportation.

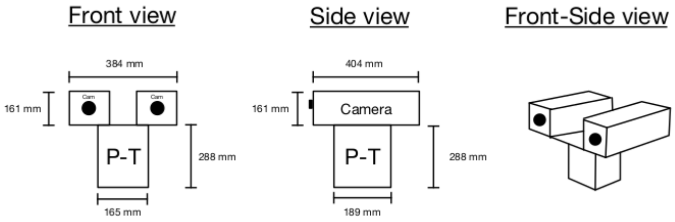


Fig. B.14: The portable pole can be equipped with a camera rig containing two cameras, e.g. RGB and Thermal camera, and a pan-tilt motor to ease view-angle adjustments.

To raise the assembled lattice mast, a steel wire is attached to the mast and directed towards a temporary installed vertical steel mast on the base platform. On this temporary installed steel mast, a manual winch system is installed, which by the use of hand-power can lift the lattice mast to its operational position where it is locked. Afterwards the temporary equipment is removed and the last tiles slot is equipped with tiles finalized the deployment of the portable pole. When deployed, 11 tiles are installed in each slot, providing a total weight of 1100 kg in the base framework.

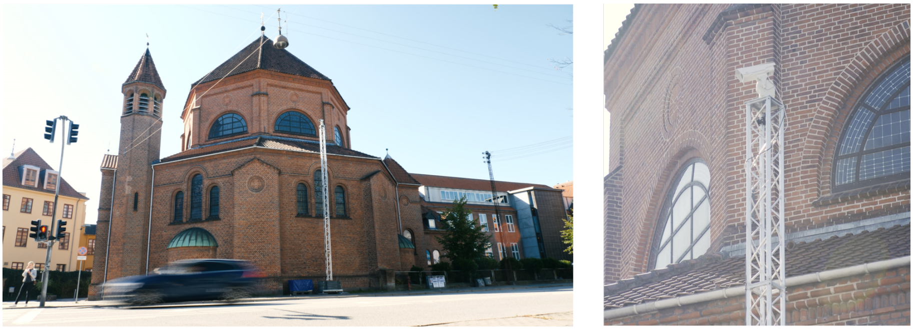


Fig. B.15: The portable pole deployed at traffic intersection. The pan-tilt motor with one RGB camera is installed on the top of the pole, which makes the RGB camera adjustable remotely.

The cameras used to derive the payload for this proposal are defined in the requirements seen in Table B.1. In addition to those cameras, we propose to include the Axis YP3040 Pan-Tilt Motor, as remote camera control has been found desirable for the setup. This, however, increases the minimum required payload weight for the portable pole with 4.2kg.

Table B.10: Summary of techincal parameteres of the TRG-pole.

Operational height [m]	Payload [kg]	Operational base dimensions [L x W x H [m]]	Transport dimensions [L x W x H [m]]	Operational Weight [Kg]	Configurable	Deployment equipment
10	12	1.2 x 1.2 x 10	-	1239	Yes	Van, trailer

The Axis YP3040 has a maximum load of 8kg meaning that a custom mounting rig needs to be created to hold both cameras whilst being mounted. The custom mounting rig is seen in Figure B.14. The overall weight of the camera setup, including a buffer, is therefore estimated to be 12 kg.

The final proposal of the TRG-pole in operation mode can be seen in Figure B.15, where you could install your equipment on, e.g. the custom mounting rig. The deployment of the portable pole is 2 hours for 2 persons and requires a van and a trailer. A visual introduction and description of the portable pole can be seen at `https://www.youtube.com/watch?v=SjZlWb3hmBo`. In Table B.10 the specifications of the TRG-pole are summarized.

5 TRAFFIC ANALYSIS USING TRG-POLE

The TRG-pole can be deployed in rural areas, which can be of particular use as there in some scenarios are no to limited existing infrastructure (light poles, balconies, trees, etc.) to mount the camera equipment in. For instance, it has been used for a traffic safety analysis as seen in Figure B.16, where there were otherwise limited options besides deploying the TRG-pole.

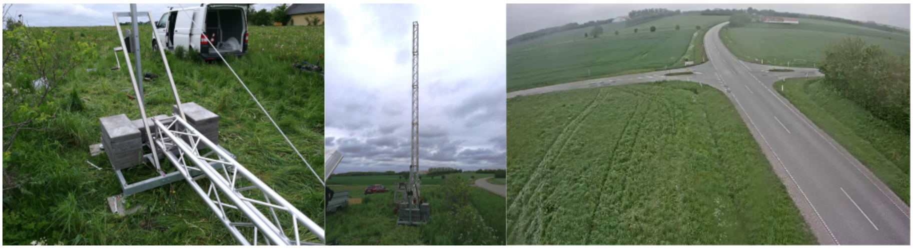


Fig. B.16: The TRG-pole is deployed at a traffic intersection with limited existing infrastructure.

But what really makes the TRG-pole a great tool, is that the very compact base frame-work allows it to be deployed in most urban areas as well. Though there might exists multiple options in most urban areas, it is however not guaranteed that it provides an ideal capturing angle for the camera equipment. A limited or bad camera view-angle will im-pact the overall quality of the traffic analysis. An example of this is shown in Figure B.17, where a traffic intersection in Aalborg is used for a traffic analysis study. The left red circle marks a camera mounted in the existing infrastructure, i.e. lighting pole, and the right red circle marks the camera installed in the TRG-pole.

The corresponding output camera feeds are seen in Figure B.18, where the existing infrastructure clearly captures the same objects as the TRG-pole does. The camera installed in the existing infrastructure do however not capture the entire cycling box and the camera's view of field do only allow a limited area of the cycling road after the cyclists begin turning right. Though the TRG-pole is deployed only a few meters away from the lighting pole, the TRG-pole provides a better capturing view for examining the potential conflicts between a cyclist and a right-turning vehicle.

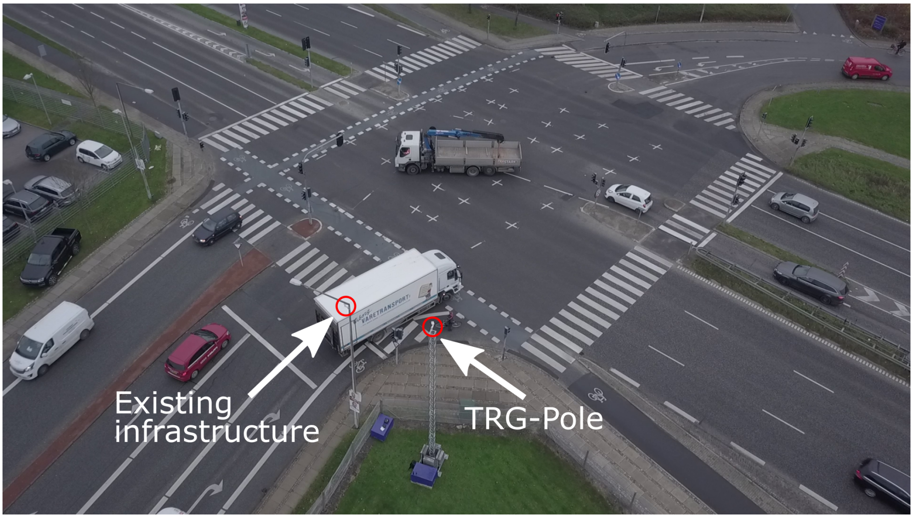


Fig. B.17: The existing infrastructure does not already provide ideal capturing positions for a traffic analysis. The usage of TRG-pole provides more ideal options due to its compactness.

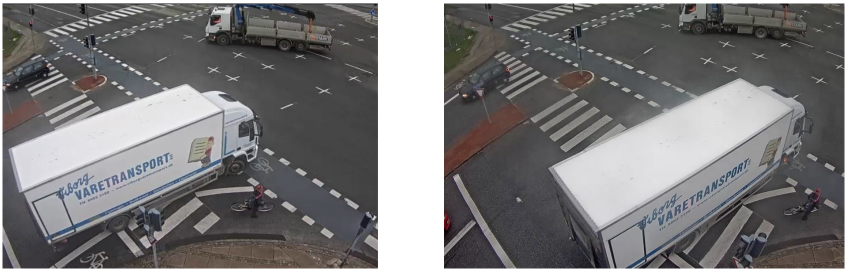


Fig. B.18: Video feed from the camera installed in (A) the existing infrastructure (B) the TRG-pole.

Using semi-automated image processing tools, e.g. RUBA [21], you can use the TRG-pole to conduct traffic analysis with a large variety of scopes, e.g. traffic counts, speed estimations, conflicts, etc. The 10 meters high pole makes a great platform for doing traffic counts as video from such a height is less occlusion prone compared to most existing infrastructure. An example of traffic counts done using the TRG-pole together with RUBA is seen in Figure B.19 , where two detectors were made to register the traffic volumes for respectively one of the entrances to the intersection (A) and one of the left-turning streams from the main road to the side road (B).

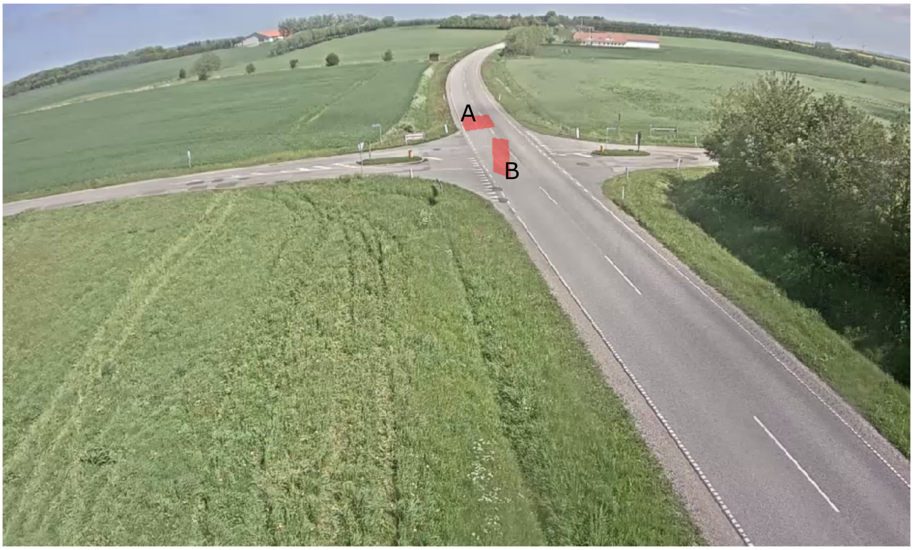


Fig. B.19: Two movement detectors registered the traffic flows through the detectors.

6 DISCUSSION

The presented portable poles types and corresponding solution have been heavily compared and discussed in Section 3.6 in a structured manner given a set of minimum requirements. The requirements have been heuristically derived and the essential requirements defined are thus biased. The remarks made for each of the surveyed solutions is therefore application depended and might still serve beneficial for other application. Most of the type-1 solution, e.g. the Miovision Scout, might be ideal to make a preliminary study at a point of interest prior to deploying a larger solution. To ensure that the final traffic analysis of the point of interest remains of high quality, the captured data must be of equally good and stable quality. A larger solution is thus necessary to ensure this during longer capturing periods due to various real-life challenges, e.g. weather, vandalism, etc.

The proposed portable pole design is not as easy deployable as most of the type-1 solutions and type-2 solutions, but do not require any guying system for maintaining and ensuring stability. In this proposal it is at most needed as a safety precaution during deployment. The main drawback of the type-4 solution is the transportation weight, which in this hybrid version of type-2 solutions and type-4 solution is reduced while remaining stable during operation. Even though the transportation weight is reduced significantly by removing the tiles from the portable pole base, the frame remains large and made out of steel, meaning that 2 persons and some deployment equipment are still required. An additional drawback of the type-4 solution, and possibly portable poles in general, is the fact they might ruin the naturalistic environment for the drivers, and therefore ruin the desired naturalistic data. The portable pole proposed in this paper do still struggle with this issue, as a portable pole looking similar to the illustration seen in Figure B.15 might still be considered obtrusive in a traffic intersection. But compared to most of the other solution, it is however considered less obtrusive.

The proposed portable pole does to some extent get inspiration and some ideas from the Trivector mobile mast, UTRaCAR, and the DLR platform solutions. These are however all considered to be quite expensive solutions, especially the Trivector mobile mast and UTRaCAR is considered expensive due to the large acquiring and remodeling price of a trailer and a car, respectively. The proposed portable pole is considered a lot cheaper to manufacture due to its simple structure and base framework.

7 CONCLUSION

This paper presents a survey, proposal, and analysis of portable poles in relation to capturing data in traffic intersection. The surveyed portable pole solutions were split into 4 general types. The type-4 solution appears to fit the defined minimum requirements most, however with a major shortcoming as it is also the lesser mobile and portable pole solution. This leads to the conclusion that for capturing the most stable and useful data, the setup must comprise the lightweight and easy mobility requirements. For the type-1 and type-2 solutions to work, various guying system must be installed on existing infrastructure to fixate the pole. If one involves the existing infrastructure, a better result would be reached if the capturing rig is mounted on the infrastructure rather than using a lightweight or compact portable pole with guying installation. The DLR solution in type-4 is considered to be the best portable pole solution based on vandalism prevention, robustness, stability, and still somehow transportable.

The DLR solution does however not completely fulfill the overall portable pole goal defined in this journal due to the limited mobility. We therefore propose a new portable pole design which combines elements from the type-2 solutions and the type-4 solution so the overall portable pole goal is reached. The proposed portable pole will get the mobility from the type-2 solutions and get the robustness and stability from the type-4 solution. The proposed design is inspired by the type-4 solution from DLR as we also propose to split usage of the portable pole into a transportation stage and an operation stage. The weight of the entire setup can dynamically and with ease be adjusted allowing a more lightweight solution and easier transportation stage.

The weight during operation is, however, still intact, such the stability is kept. The proposed portable pole can be deployed by 2 persons in 2 hours in both rural areas as well as urban areas due to its compactness.

ACKNOWLEDGEMENTS

The authors would like to thank Kay Gimm and Sascha Knake-Langhorst from Deutsches Zentrum für Luft- und Raumfahrt(DLR) for the aid in information about the Test field AIM (Application Platform f or Intelligent Mobility) and the UTRaCar.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] European Commission, *2017 road safety statistics: What is behind the figures? - Fact Sheet*. European Commission, 2018. [Online]. Available: http://europa.eu/rapid/press-release_MEMO-18-2762_en.pdf

[2] European Commission, *Towards a European Road Safety Area: Policy Orientations on Road Safety 2011-2020*. European Commission, 2010. [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/road_safety/pdf/com_20072010_en.pdf

[3] V. L. Neale, T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman, "An overview of the 100-car naturalistic study and findings," *National Highway Traffic Safety Administration, Paper*, vol. 5, p. 0400, 2005.

[4] G. Davis and J. Hourdos, "Development of analysis methods using recent data: Shrp2 safety research," *Transportation Research Board of the National Academies, Tech. Rep*, 2012.

[5] M. P. Philipsen, M. B. Jensen, R. K. Satzoda, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, "Day and night-time drive analysis using stereo vision for naturalistic driving studies," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1226–1231.

[6] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *Bulletin Lund Institute of Technology, Department*, no. 70, 1987.

[7] A. Fyhri, H. Sundfør, T. Bjørnskau, and A. Laureshyn, "Safety in numbers for cyclists—conclusions from a multidisciplinary study of seasonal change in interplay and conflicts," *Accident Analysis & Prevention*, vol. 105, pp. 124–133, 2017.

[8] T. K. O. Madsen and H. Lahrmann, "Comparison of five bicycle facility designs in signalized intersections using traffic conflict studies," *Transportation research part F: traffic psychology and behaviour*, vol. 46, pp. 438–450, 2017.

[9] L. Sakshaug, A. Laureshyn, Å. Svensson, and C. Hydén, "Cyclists in roundabouts—different design solutions," *Accident Analysis & Prevention*, vol. 42, no. 4, pp. 1338–1351, 2010.

[10] Transport-, Bygnings- og Boligministeriet. Bekendtgørelse om køretøjers største bredde, længde, højde, vægt og akseltryk (dimensionsbekendtgørelsen). [Online]. Available: https://www.retsinformation.dk/pdfPrint.aspx?id=137554

[11] International Transport Forum. Permissible maximum dimensions of lorries in europe. [Online]. Available: https://www.itf-oecd.org/sites/default/files/docs/dimensions_0.pdf

[12] S. Plainis, I. Murray, and I. Pallikaris, "Road traffic casualties: understanding the night-time death toll," *Injury Prevention*, vol. 12, no. 2, pp. 125–138, 2006.

[13] Miovision. Scout video collection unit. [Online]. Available: https://miovision.com/scout/

[14] C. Masts. Clark masts ft series. [Online]. Available: http://www.clarkmasts.com/media/dyn-docs/products/ft-masts-brochure.pdf

[15] C. Masts. Clark masts qt series. [Online]. Available: http://www.clarkmasts.com/products/telescopic-masts/qt-series/

[16] Litec Strutture & Soluzioni. Flyintower 7.5-500 catalogue. [Online]. Available: http://www.litectruss.com/Litec/media/litec/Downloads/FLYINTOWER_7-5-500_catalogue.pdf?ext=.pdf

[17] Dlr - institut für verkehrssystemtechnik - utracar und momocar. [Online]. Available: http://www.dlr.de/ts/desktopdefault.aspx/tabid-1237/5441_read-12153/

[18] M. Junghans, "Situations- und gefahrenerkennung in verkehrsszenen," in *Kolloquium Verkehrsmanagement und Verkehrstelematik*, Dresden, Germany, May. 8 2013. [Online]. Available: http://www.vimos.org/cms/data/uploads/termine/junghans_sgv.pdf

[19] C. Masts. Clark masts model 804/15-6 heavy duty trailer mast. [Online]. Available: http://www.clarkmasts.com/media/dyn-docs/products/model-804-15-6-trailer-mast-cat-no-19877.pdf

[20] D. I. of Transportation Systems, “Aim mobile traffic acquisition: Instrument toolbox for detection and assessment of traffic behavior,” *Journal of large-scale research facilities*, no. 2, A74, 2016.

[21] C. H. Bahnsen, T. K. O. Madsen, M. B. Jensen, H. S. Lahrmann, and T. B. Moeslund, “The ruba watchdog video analysis tool,” 2018.

Paper C

Multi-view Traffic Intersection Dataset: Performance Analysis and Comparison

Morten B. Jensen and Thomas B. Moeslund

This paper is ongoing work
2018

1 INTRODUCTION

From a computer vision point of view, the traffic surveillance domain is a very challenging area due to the large amount of diverse scenes, objects and in particular objects overlapping each other generating occlusion. Occlusion can due to humans cognitive abilities be trivial, but for a computer to reach the same level, we need to make it quite intelligent by showing it a lot of diverse training images.

The easiest way to prevent occlusion is to consider the placement of your sensors, which are usually placed on either existing infrastructure, e.g. traffic light poles, or mounted on some portable pole [1]. These options provide a side-top view-angle, which is usually fine, but might be challenged with large objects, e.g. trucks and buses, occluding large areas in the scene. A natural solution raised by the traffic researchers is to use a drone, which would provide perfect bird view and thus solve all above issues. This might be true for the case of large objects such as vehicles, trucks, and buses, but distinguishing between some vulnerable road users, e.g. pedestrians and cyclists, might become more difficult from a 50 meter bird view-angle. An illustration of 3 popular approaches for capturing data at traffic intersections are seen in Figure C.1. The portable pole can in some scenarios provide a better view-angle in case the existing infrastructure options are limited.

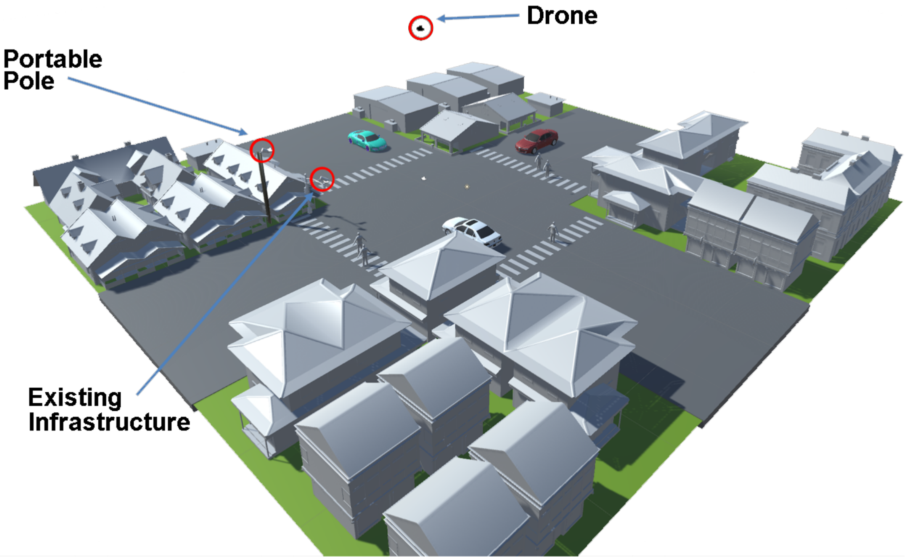


Fig. C.1: A traffic intersection is equipped with 3 different capturing angels by mounting a camera in existing infrastructure, deploying a portable pole, and utilizing a drone equipped with a camera.

Making the computer intelligent and able to reduce the dataset to only interesting time sequences usually refer to the computer vision stages: object detection, classification and tracking. In this paper, we will only look into object detection. Object detection has been a large computer vision area for ages and has to some extend been solved with simple algorithms such as the Viola-Jones framework [2].The Haar-like features were used for quite some time until Histogram of Oriented Gradients (HOG) was introduced, which when combined with Support Vector Machine (SVM) outperforms the Viola-Jones framework [3]. Additionally, HOG+SVM has in newer updated versions been able to perform very well with a rather small selection of training images due to inclusion of newer optimization methods.

Machine learning has seen a large boost and is used in almost every recent publication in major journals and conferences. There are multiple deep learning object detectors, such as Mask R-CNN [4], Single Shot MultiBox Detector (SSD) [5] and You-Only-Look-Once (YOLO) [6], which have all seen great use and outperformed previous entries various detection challenges, e.g. Pascal VOC [7], COCO [8], VIVA-Challenge [9], and VIRAT [10].

In this paper, we collect a dataset from 2 different view-angles, namely view from existing infrastructure and a drone, which to some extend is visualized in Figure C.1. To compare the view-angles, we annotate the dataset with both bounding box and instance segmentation annotations. We apply the state-of-the-art object detector Mask R-CNN on the dataset with the purpose of comparing the performance on different objects in different view-angles. The contributions of this paper is thus twofold:

- Collecting, annotating, and publishing of a multi-view traffic intersection dataset.
- Conducting a comparative study using state-of-the-art method between the two view-angles.

2 CAPTURING TRAFFIC SURVEILLANCE DATA: CHALLENGES

The current widespread solution for recording traffic video data at traffic intersections is by the use of cameras mounted in existing infrastructure, e.g. poles, as seen in Figure C.2. As mentioned in the introduction, this view-angle is however not always ideal as larger objects, e.g. busses, trucks, etc, occludes smaller objects as illustrated in Figure C.3. This occlusion is a quite well known problem, which could potentially be partly solved by deploying a portable pole providing a higher capturing height [1].

But in order really to get the full overview of a traffic intersection a drone is an obvious choice given its bird-view as illustrated in Figure C.4.

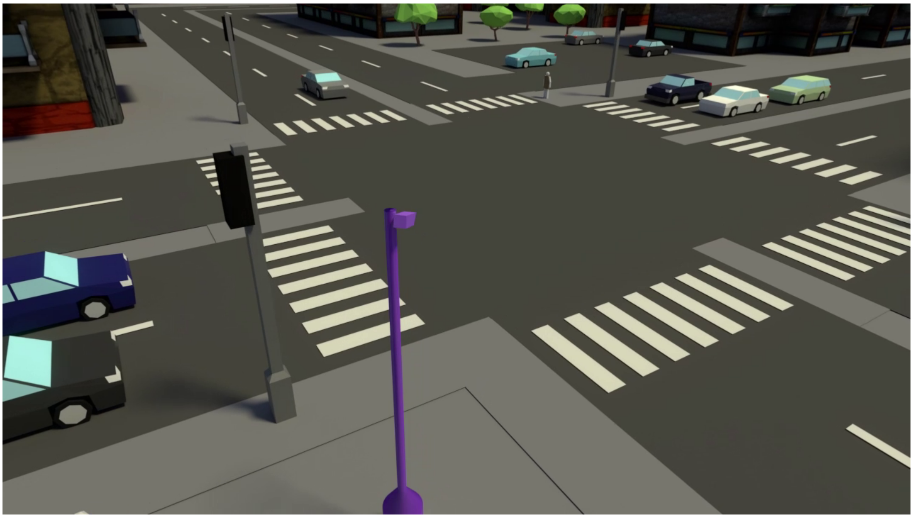
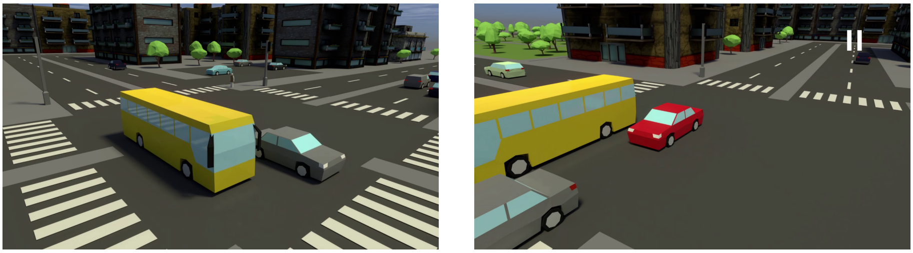


Fig. C.2: Camera mounted in existing infrastructure at a traffic intersection.



(a)

(b)

Fig. C.3: An illustration of the output from a camera mounted in existing infrasruture is seen in (a). Given this view-angle, the bus and other large objects clearly introduce an occlusion problem as illustrated in (b).

Though the drone view-angle clearly provides a better overview and lesser degree of occluding objects in the traffic scene. It is however quite difficult to see smaller objects, e.g. cyclists and pedestrians. This is illustrated in Figure C.5. The capturing height could of course be smaller, but currently the regulations of drones is quite strict which defines some restrictions on these possibilities.

Choosing between the two different view-angles is a very application-based decision. The drone can provide you with data very rapidly without any major setup, it do however in Denmark require a permission from the local police as well as a licensed pilot to do so. Though the drone provides you with a very nice overview of most traffic scenes, traditional drones do

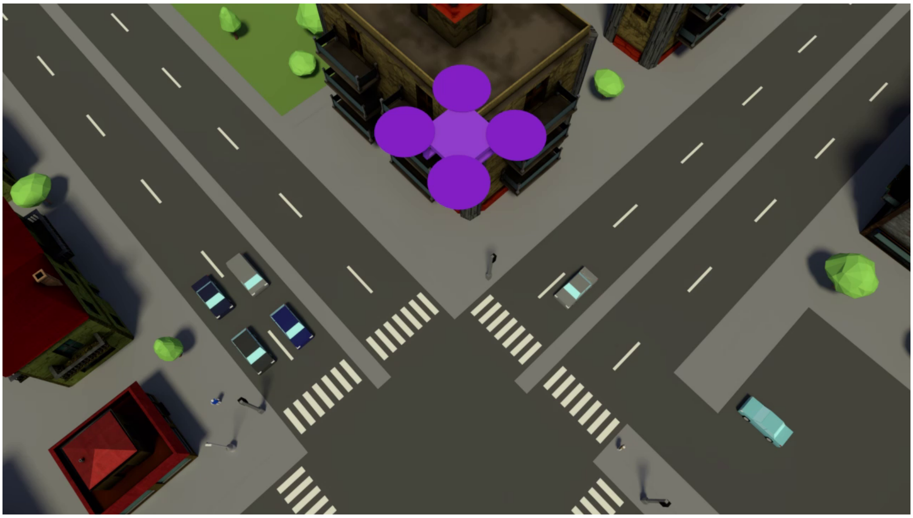
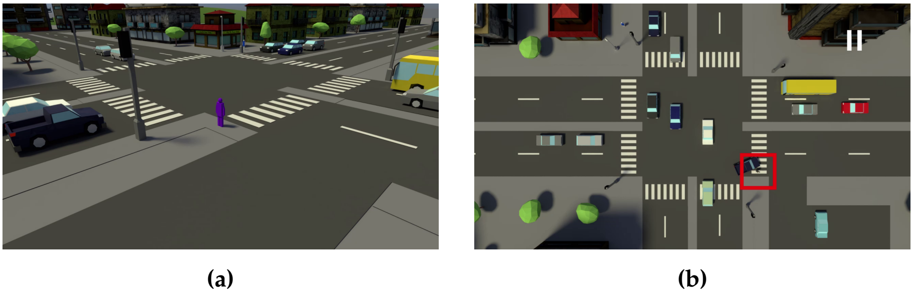


Fig. C.4: A drone hovering above a traffic intersection.



(a)

(b)

Fig. C.5: (a) A pedestrian ready to cross the intersection. (b) It is very hard to see and thus detect the pedestrian in the drone view-angle.

however not allow for capturing continuously for a very long period of time. Most batteries in high-end consumer product allow up to 24-30 minutes of flight time. For some applications and pilot test, this might be sufficient, but for more comprehensive studies, longer recordings are needed. Additional batteries scales the recording time linearly, but requires the drone to land and take-off again. If we consider more industrial drones, these allow a power cable attached to it, which to some extend make capturing continuously possible. This do however put the electrical motors on overload.

Though some expensive industrial drones can remain operational during rainy and windy conditions, most consumer-friendly drones needs to be taken down if the wind speed exceeds 10 m/s or if it starts raining.

Table C.1: Overview of the dataset.

View-angle	Resolution	Bicycle	Car	Bus	Lorry
Infrastructure	640x480	2003	9989	2113	4778
Drone	1920x1080	2909	32550	3607	11208

3 DATASET

As mentioned in the introduction, the datasets consists of two different view-angles: Existing infrastructure and a drone. The dataset consists of 3100 synchronized and annotated frames from each of the two view-angles. Each frame is annotated with both an axis-aligned bounding box as well as on a pixel level allowing instance segmentation. The annotation scheme is following the COCO format and classes, which in this case provides 4 different annotations in the dataset, namely: bicycle, car, bus and lorry. An overview of the specifications of the dataset are seen in Table C.1. The infrastructure view-angle is captured using an AXIS M1124-E camera with a VGA resolution and 30 FPS. The drone view-angle is captured using a DJI Mavic Pro Drone and its standard installed camera. The drone view-angle is captured in full HD resolution and 30 FPS. The dataset is captured in an intersection in downtown Aalborg, Denmark.

The annotation distribution in the dataset for the infrastructure and drone view-angle are seen in Figure C.6 and Figure C.7, respectively.

4 EVALUATION

For evaluating the two view-angles, we applied the Mask R-CNN [4] for object detection. We used a model pre-trained on the Microsoft COCO dataset and applied that on dataset. Given an overlap criterion of 50 %, the Mask R-CNN achieved a mean average precision (mAp) of 22.62 % for infrastructure and 27.75 % for drone, which is shown in the precision-recall curves in Figure C.8 and Figure C.9, respectively.

By examining Figure C.8, it is clear that neither of the 4 classes are particularly well-performing. The best performing classes is "car" with an average precision (AP) of 47.49 %, which is followed by bus and lorry on 26.94 % and 10.21 %, respectively. The Mask R-CNN object detector on the infrastructure view-angle do detect a few bicycles, but by visual inspection of the the recall on the x-axis, it not even 10 % of the total amount of bicycles present in the infrastructure dataset. If we however compare this to the performance on the drone dataset in Figure C.9, the infrastructure view-angle is in fact better as no bicycles are detected at all in the drone dataset. The 3 other classes

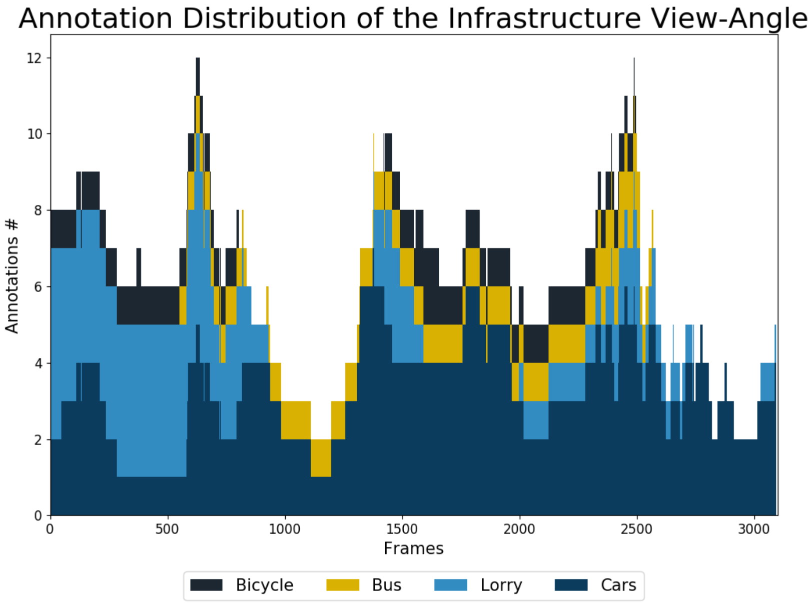


Fig. C.6: Stacked barchart of the annotation distribution from the infrastructure view-angle.

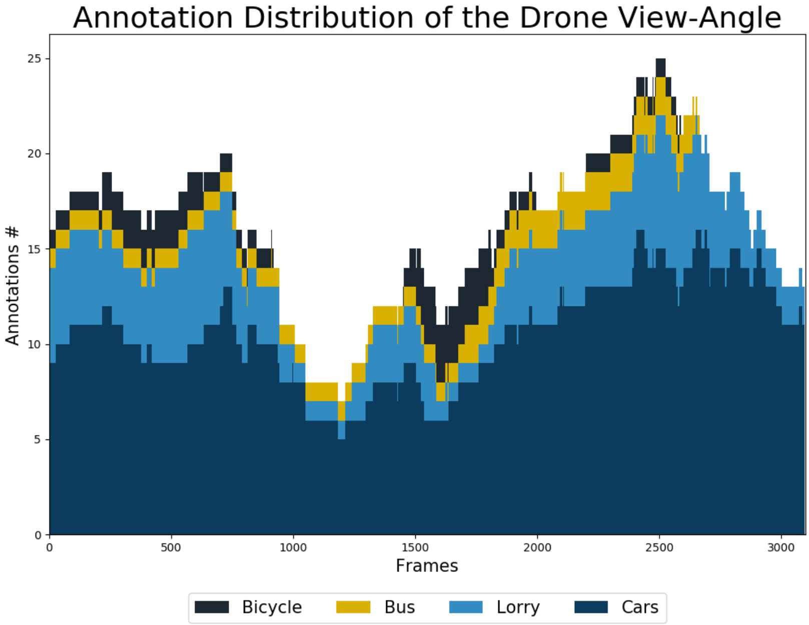


Fig. C.7: Stacked barchart of the annotation distribution from the drone view-angle.

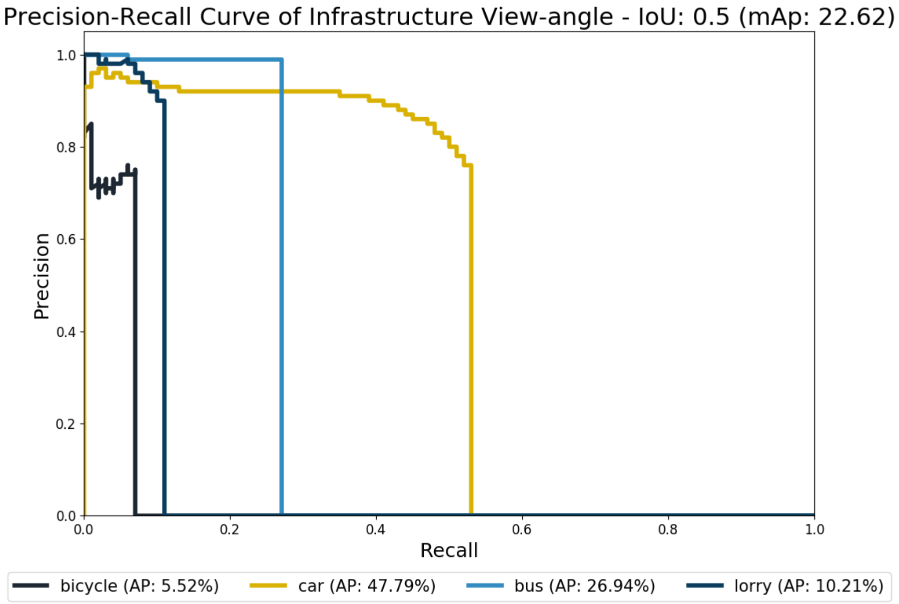


Fig. C.8: Precision-recall curve with IoU of 50 % of the infrastructure view-angle.

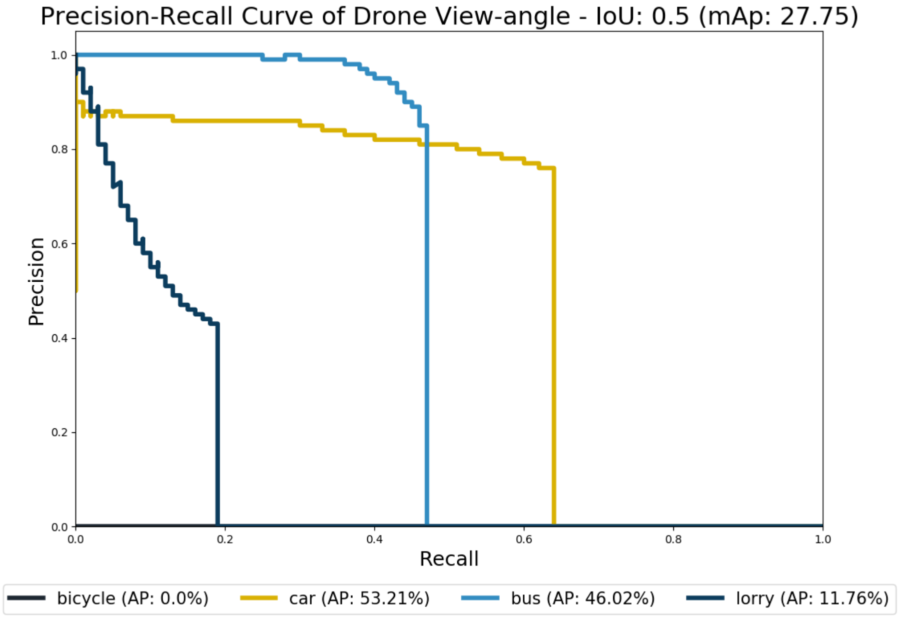


Fig. C.9: Precision-recall curve with IoU of 50 % of the drone view-angle.

are however all performing better in the drone view-angle compared to the infrastructure view-angle.

To provide a better insight into whether the bicycle class is detected in the dataset. We vary the intersection over union (IoU) or overlap criterion from 0.01, corresponding to allowing a very small overlap between detection and ground truth to be considered at true positive, to 1.00, meaning a perfect detection. As we vary the IoU we measure the AP and mAP for all the classes, which is shown in Figure C.11 and Figure C.10 for infrastructure view-angle and drone view-angle, respectively.

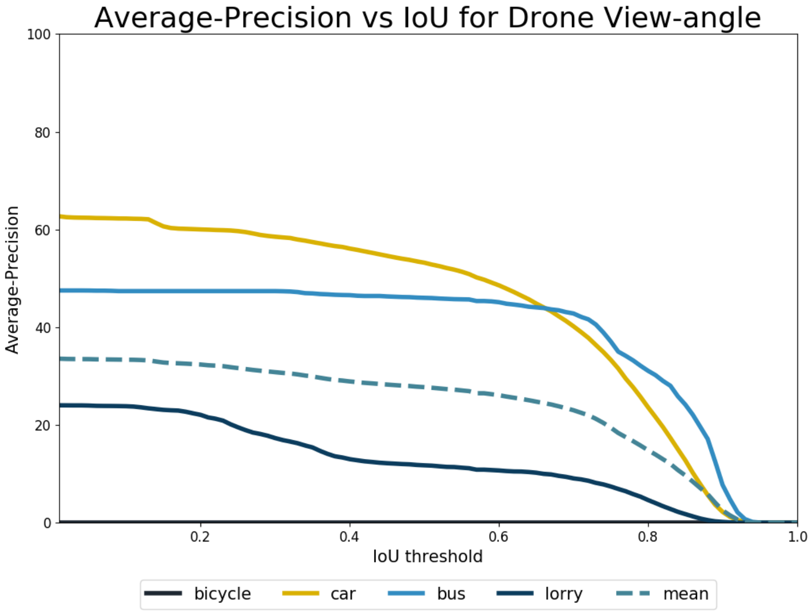


Fig. C.10: The AP as a result of varying the IoU criterion in the drone view-angle.

By examining and comparing the results presented in Figure C.11 and Figure C.10, it is clear that even if we allow for detections that barely overlap the ground truths annotations, we still do not detect any bicycles in the drone view-angle. Nor does it significantly improve the AP of the bicycles in the infrastructure view-angle.

This ultimately stress that the drone view-angle might provide a better overview from a pure visual inspection. But if we want to apply pre-trained deep learning models on it, the current available dataset used for training these models is not sufficient.

In Figure C.12 and Figure C.13 samples of the annotated data are presented.

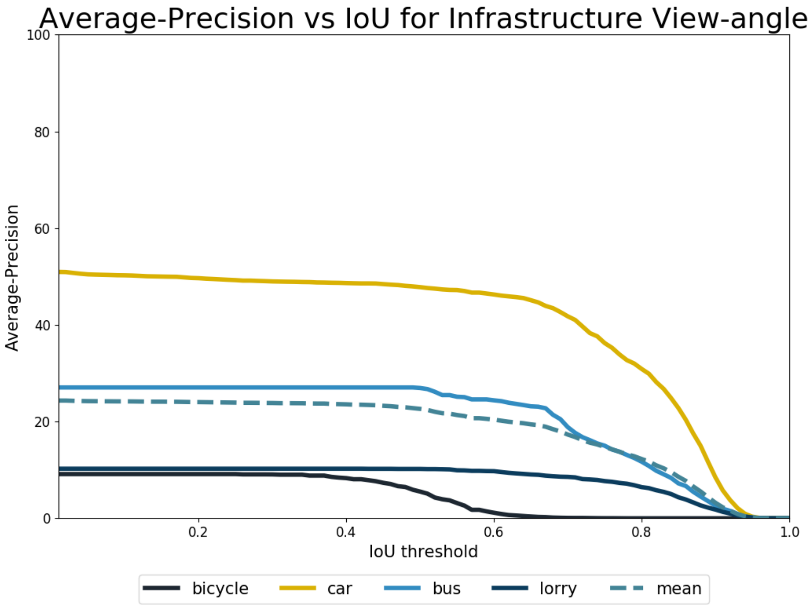
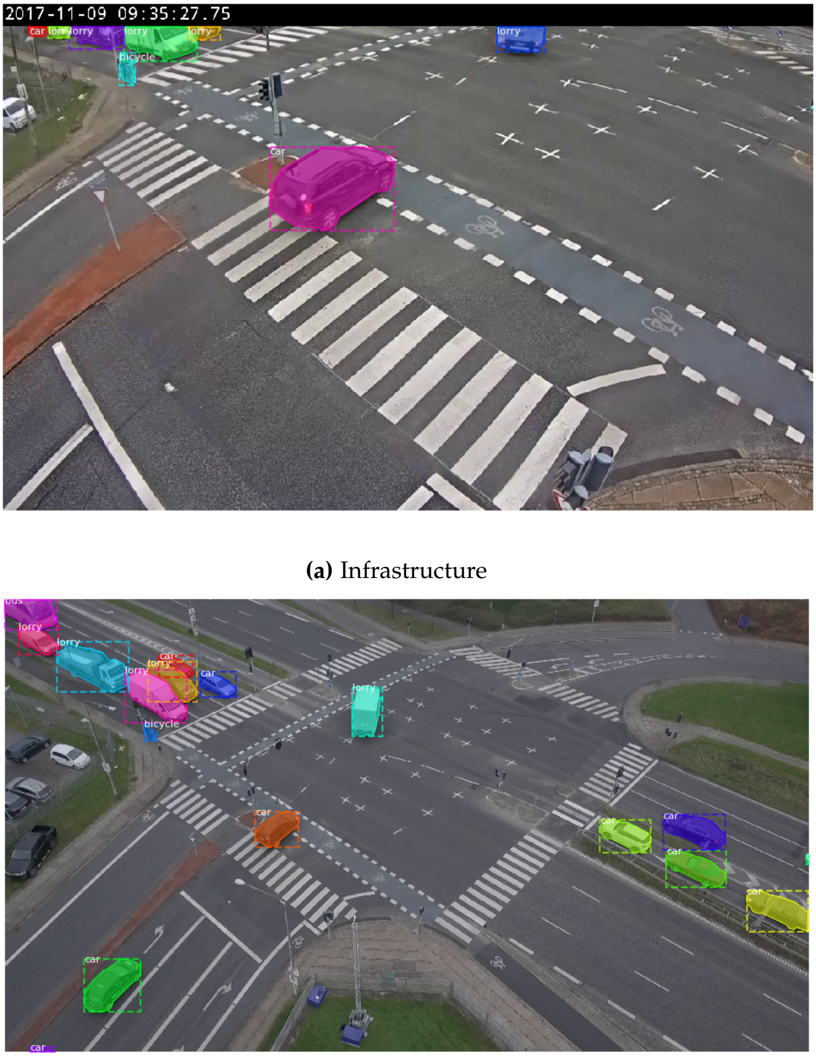


Fig. C.11: The AP as a result of varying the IoU criterion in the infrastructure view-angle.

5 FUTURE WORK

As this paper is still ongoing work, a lot of future investigations and work has to be carried out. These investigations and work are:

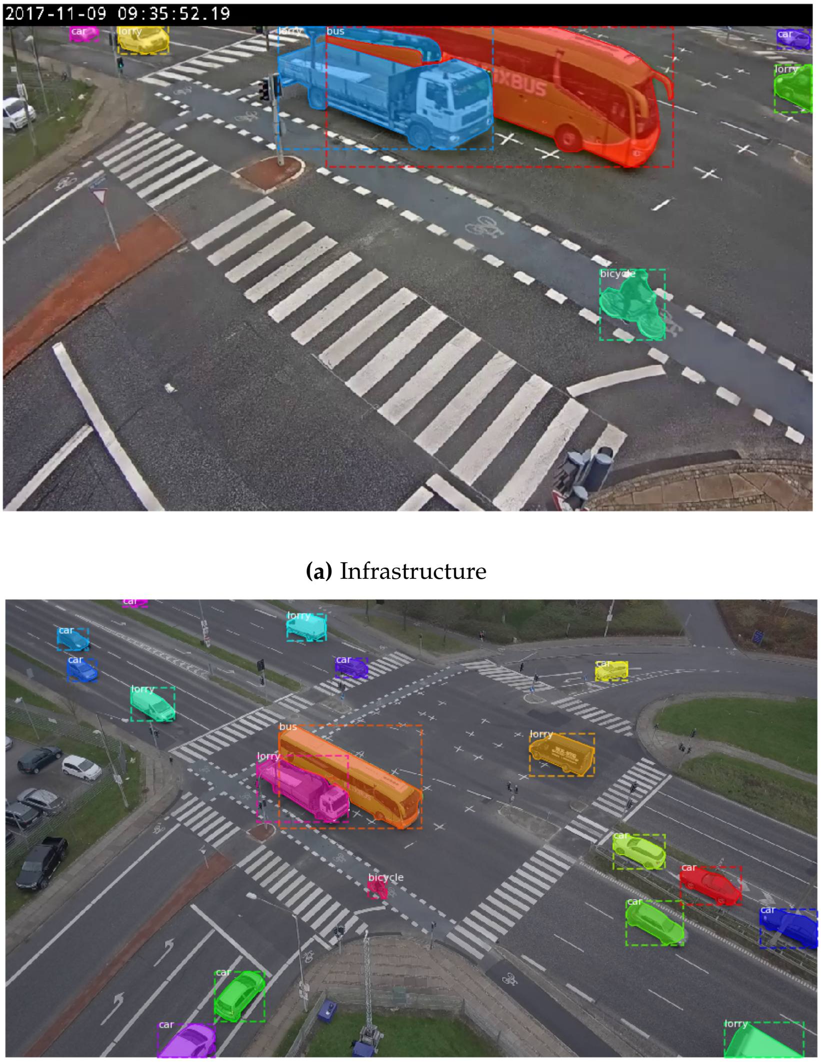
- Related work of both object detection from UAV but also provide an overview of relevant available dataset and their short comings.
- The annotated dataset allow for instance segmentation, which Mask R-CNN also do. Investigate and evaluate the different view-angles performance on a pixel-level.
- Utilize more detectors.
- Train own models with varying amount of training data to see how the performance changes.
- Looking into using synthetic data to apply fine-tune the pre-trained models in the unfamiliar view-angles.
- Capture more data with multi-modal (RGB+Thermal) as well as multi-view.
- And as always... Annotate more data.



(a) Infrastructure

(b) Drone

Fig. C.12: Annotated frame 26 of each of view-angle.



(a) Infrastructure

(b) Drone

Fig. C.13: Annotated frame 759 of each of view-angle.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. B. Jensen, C. H. Bahnsen, H. S. Lahrmann, T. K. O. Madsen, and T. B. Moeslund, "Collecting traffic video data using portable poles: Survey, proposal, and analysis," *Journal of Transportation Technologies*, vol. 8, no. 4, 2018.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," vol. 1, pp. I–I–518, 2001.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[4] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. Cham: Springer International Publishing, 2016, pp. 21–37.

[6] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan 2015.

[8] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[9] Laboratory for Intelligent and Safe Automobiles, UC San Diego. (2015) Vision for Intelligent Vehicles and Applications (VIVA) Challenge. http://cvrr.ucsd.edu/vivachallenge.

[10] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, June 2011, pp. 3153–3160.

Part III

Object Detection

Paper D

Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives

Morten B. Jensen, Mark P. Philipsen, Andreas Møgelmose,
Thomas B Moeslund, and Mohan M Trivedi

The paper has been published in the
IEEE Transactions on Intelligent Transportation Systems Vol. 17(7),
pp. 1800–1815, 2016.

Please note that the above article is updated slightly. The updates are:

- An additional reference has been included [1].
- 106 citations to this reference are added.

The additional reference is a Master's thesis: Mark P. Philipsen & Morten B. Jensen, *Computer Vision at Intersections: Explorations in Driver Assistance Systems and Data Reduction for Naturalistic Driving Studies*, Master's thesis, Aalborg University, Denmark, 2015.

There is an overlap between the content of that thesis and the content of the article. The citations are added to make this overlap evident.

ABSTRACT

This paper presents the challenges that researchers must overcome in traffic light recognition (TLR) research and provides an overview of ongoing work. "The aim is to elucidate which areas have been thoroughly researched and which have not, thereby uncovering opportunities for further improvement. An overview of the applied methods and noteworthy contributions from a wide range of recent papers is presented, along with the corresponding evaluation results. The evaluation of TLR systems is studied and discussed in depth, and we propose a common evaluation procedure, which will strengthen evaluation and ease comparison. To provide a shared basis for comparing TLR systems, we publish an extensive public dataset based on footage from US roads. The dataset contains annotated video sequences, captured under varying light and weather conditions using a stereo camera." [1] The dataset, with it's variety, size, and continuous sequences should challenge current and future TLR systems.

1 INTRODUCTION

"The efficiency of transportation systems fundamentally affect the mobility of the workforce, the environment, and energy consumption, which in turn dictates foreign policy. Since transportation is a major part of people's lives, their health and well-being is directly related to it's efficiency, safety, and cleanliness. Many future improvements to transportation systems will come from innovations in sensing, communication, and processing [2, 3]." [1]

"The automobile revolution in the early 20th century led to a massive increase in road transportation, and the contemporary road network was incapable of handling the rapid increase in traffic load. To allow for efficient and safe transportation, traffic control devices (TCD) were developed to guide, regulate, and warn drivers. TCDs are infrastructure elements that communicate to drivers, e.g. signs, signaling lights and pavement markings [4]. Figure D.1 shows an illustration of a road scene with some of the many TCDs." [1]

"TCDs are especially important in complex settings such as intersections, where a lot of information must be communicated. Informing drivers is a balance between providing sufficient information while avoiding to burden and distract drivers excessively. A driver's ability to obtain information from TCDs is limited by the amount of information and the time available to comprehend the information. High speed and overwhelming amounts of information may hence lead to errors from oversights and stress [4]. For TCDs to function properly, all road users are required to abide, otherwise dangerous situations occur. Drivers sometimes purposely disregarded TCDs. One study shows that more than 1/3 of Americans admit to having purposefully run a red light during the past month [5]. In many cases, failure to comply is unintentional and caused by e.g. speeding to make it through the intersection in time, aggressive

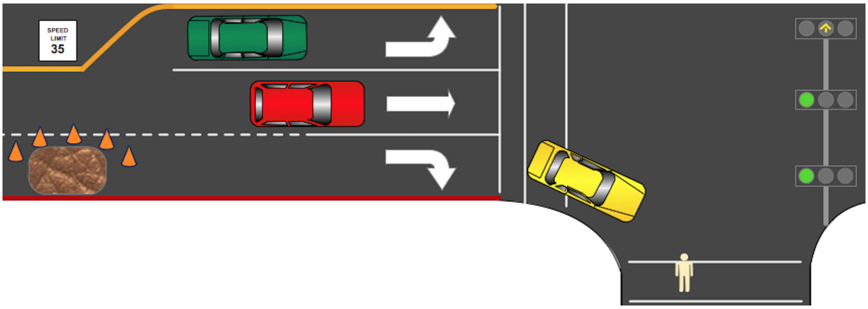


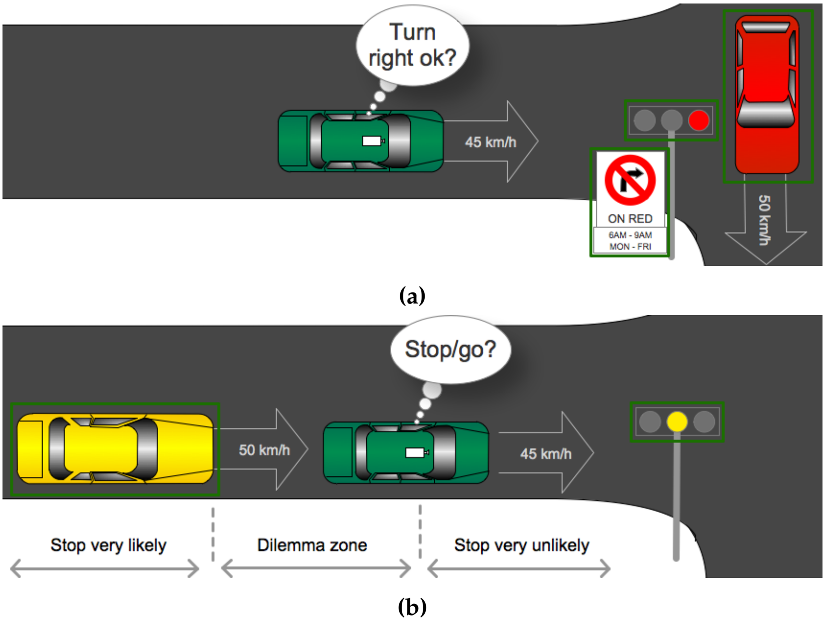
Fig. D.1: *"Traffic control devices for safe and efficient traffic flow."* [1]

driving by closely following the car in front, distraction [6], misunderstandings, or faulty TCDs." [1]

"The complex task of driving is easy, most of the time," [1] since many driving sub-tasks are automated. Effortless driving results in unfocused drivers to whom critical events can be perceived with an added delay. "Stressful driving where the driver is very focused and attentive can delayed reaction time, because of fatigue, and mental overload [7]." [1]

Widespread autonomous driving lies years in the future, in the meantime lives can be saved by having driver assistance systems (DAS) monitor the environment and warn or intervene in critical situations. For DAS to best support the driver, it must attempt to make up for the driver's deficiencies. *"An example of a driver deficiency is noticing and recognizing TCDs. Studies show that drivers notice some TCDs better than other; speed limit signs are almost always noticed, while pedestrian crossings signs are mostly overlooked [8]."* [1]

"For all parts of DAS, the urban environment possesses a wealth of challenges, especially to systems that rely on computer vision. An important challenge is recognizing TLs at intersections. In 2012, 683 people died and 133,000 people were injured in crashes that involved red light running in the USA [9]. Ideally, TLs should be able to communicate both visually and using infrastructure to vehicle (I2V) by means of radio communication. Introducing I2V on a large scale requires substantial investments in infrastructure, which are unlikely in the near future. Intersections are some of the most complex challenges that drivers encounter, making visual recognition of TLs an integral part of DAS. The yellow light dilemma is one example where DAS can support drivers. When entering an intersection with a yellow TL, the driver must make a decision of whether to stop or to keep going and cross the intersection. The interval where this decision is difficult for most people is in the range of 2.5-5.5 seconds before entering the intersection [10]. Outside this interval the decision is typically quite clear. The reaction times of drivers is longest in the center of the interval, where the decision is the most difficult. Figure D.2 shows two scenarios where information from different sensors and intelligent systems can be combined



(a)

(b)

Fig. D.2: *"Fused DAS system in intersection scenarios. (a) Turn right on red assistance. (b) Dilemma zone assistance."* [1]

and provide driver assistance." [1]

Until now no comprehensive survey of traffic light recognition (TLR) research has been published. The sub-problem of traffic light detection has been survey in [11] and in [12] we presented an introductory overview of ongoing work on traffic light detection along with the LISA Traffic Light Dataset. Most published TLR systems are evaluated on datasets which are unavailable to the public. *"This makes comparison between existing methods and new contributions difficult."* [1] The contributions made in this survey paper are thus, fourfold:

1. Clarifying challenges facing TLR systems.
2. Overview of current methods in TLR research.
3. Common evaluation procedure for TLR systems.
4. High resolution, annotated, stereo video dataset.

"The paper is organized as follows: Section 2 touches on computer vision areas similar to TLR. In section 3, TL appearance is discussed along with common challenges facing TLR systems. Section 4 presents the typical TLR system. Recent work is examined in section 5. In section 6 evaluation of TLR systems is reviewed, and a

common procedure is proposed. Section 7 presents the LISA TL Dataset. In section 8, experiences, and future possibilities are discussed. Section 9 rounds of with the findings made through out the survey." [1]

2 RELATED COMPUTER VISION CHALLENGES

"Before delving into the research made in TLR, it is interesting to examine the challenges, methods, and experiences from related computer vision problems which in many cases will be similar. Related computer vision problems would be: traffic sign recognition, taillight, headlight, and lane detection." [1]

"Detection of traffic signs is challenging when subject to varying lighting, viewpoints, and weather conditions. All of these issues suggest that relying solely on color is problematic, therefore shape information is useful for sign detection. An example of the use of shape information is seen in [13]. Relying on shape can also be challenging with both traffic signs and TLs, as the angle between the ego-vehicle and the sign or TL will impact the perceived shape of the object, resulting in a new shape variation. Developing robust vision based DAS that works under changing lighting, at varying distances, and under mixed weather conditions is a difficult task as stated in [14], which mentions that cross-over procedures for handling environmental transitions should be investigated." [1] Following the 2012 survey on traffic sign recognition [8], the focus has shifted entirely to learning-based traffic sign detectors *"and the problem is considered solved on a subset of signs [15, 16]."* [1] The same paradigm shift has not yet materialized in TLR.

"Most vehicle detection and brake light detection at night utilize monocular cameras and rely on the symmetry of tail and head lights for detecting vehicles as seen in [17–22]. [23] detects head and tail lights using cues from lane detection, with the purpose of automatic switching between high-beam and low-beam. Similarly, cues from lane detection are important additions to TLR systems in order to determine the relevance of TLs. A recent paper on lane detection is [24], where a context aware framework for lane detection is introduced, this can significantly reduce the required computational demand by scaling the detection algorithm based of the state of the ego-vehicle and the road context. The same paper references several comprehensive surveys on lane estimation techniques, one being [25], where work done across multiple modalities is reviewed. In [26, 27] the gaze and attention of the driver is determined. This is essential information for DAS, since it can be used to determine if the driver should be notified as e.g. in [28] where the driver is alerted and safety systems are engaged if the driver is inattentive for a prolonged period of time." [1]

3 TRAFFIC LIGHTS: CONVENTIONS, STRUCTURE, DYNAMICS, AND CHALLENGES

TLs regulate the traffic flow, by informing drivers about the right of way. Right of way is given in a manner which minimize conflicts between vehicles and pedestrians traveling incompatible paths through the intersection. *"TLs are by design made to stand out and be easily visible."* [1] Their primary components are bright colored lamps, usually circle or arrow shaped. The lamps are surrounded by a uniformly colored container. *"The most common TL configuration is the red-yellow-green light, where each state indicates whether a driver should stop, be prepared to stop, or keep driving. A variety of other TLs have been created as a result of complex intersections. Figure D.3 shows some of the allowed vertical configurations of TLs in California."* [1]

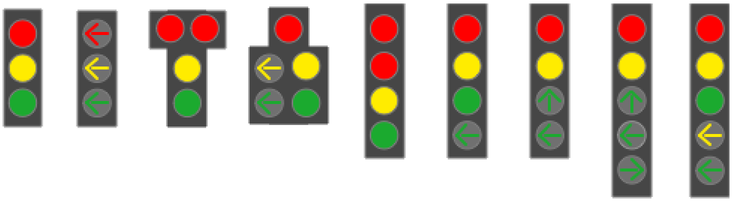
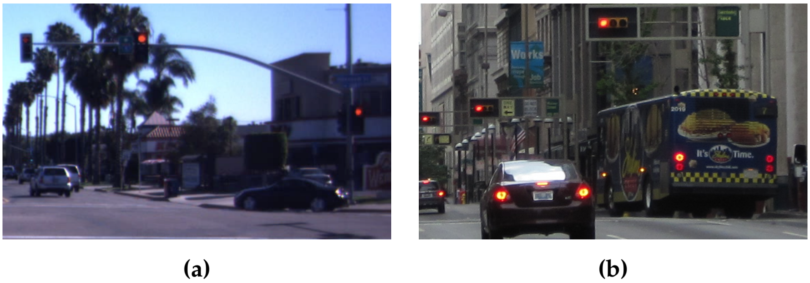


Fig. D.3: "*Examples of vertical TLs found in California. [29]*" [1]

"The orientation, color, size, and shape of the container will vary country to country and even city to city. An example of differently oriented and colored TLs within the USA is seen in Figure D.4. There are two methods for mounting TLs, suspended and supported, this is evident in Figure D.4(a). The supported variety has proven the most difficult for existing TLR systems, as discussed in subsection 3.1." [1]

"Besides the various configurations of TLs, the state sequence is an important characteristic of a TL. An example of a state sequence for the basic red-yellow-green



(a) **(b)**

Fig. D.4: *"(a) San Diego, California. (b) Cincinnati, Ohio."* [1]

light is shown in Figure D.5." [1]

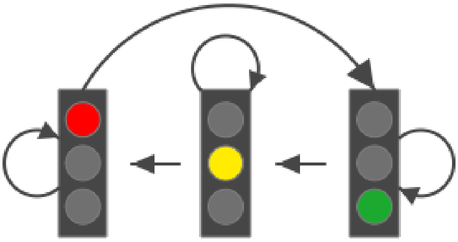


Fig. D.5: *"Basic TL sequence for states: green, yellow, and red."* [1]

"For increasing road safety and making it easier for drivers when driving across states, TLs in USA are regulated by the Federal Highway Administration in the Manual on Uniform Traffic Control Devices [30]. Most countries in Europe have signed the Vienna Convention on Road Signs and Signals [31], requiring TLs to meet a common international standard." [1]

3.1 Challenges in recognizing traffic lights

"Although TLs are made to be easily recognizable, influences from the environment and e.g. sub-optimal placement can make successful detection and recognition difficult, if not impossible. Issues include:" [1]

- *"Color tone shifting and halo disturbances [32] e.g. because of atmospheric conditions of influences from other light sources Figures D.6c, D.6d, D.6k and D.6l." [1]*
- *"Occlusion and partial occlusion because of other objects or oblique viewing angles [32]. Especially a problem with supported TLs [33–35]. Figures D.6e to D.6g."* [1]
- *"Incomplete shapes because of malfunctioning [32] or dirty lights. Figure D.6a."* [1]
- *"False positives from, brake lights, reflections, billboards [36, 37], and pedestrian crossings lights. Figure D.6h."* [1]
- *"Synchronization issues between the camera's shutter speed and TL LED's duty cycle. Figures D.6i and D.6j."* [1]

"Inconsistencies in TL lamps can be caused by dirt, defects, or the relatively slow duty cycle of the LEDs. The duty cycle is high enough for the human eye not to notice that the lights are actually blinking. Issues arise when a camera uses fast shutter speeds, leading to some frames not containing a lit TL lamp. Saturation is another aspect that can influence the appearance of the lights. With transition between day

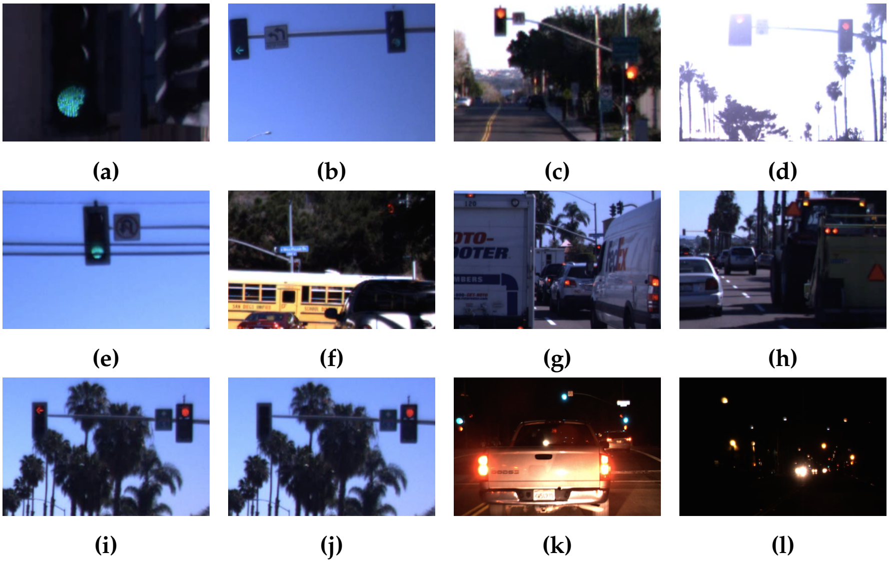


Fig. D.6: *"Examples of frames from the collected dataset."* [1]

and night, the camera parameters must be adjusted to let the optimal amount of light in and avoid under or over-saturation. [38] introduces an adaptive camera setting system, that change the shutter and gain settings based upon the luminosity of the pixels in the upper part of the image." [1]

4 TRAFFIC LIGHT RECOGNITION FOR DRIVER ASSISTANCE SYSTEMS

"Computer vision problems like TLR can be divided into three sub problems: detection, classification, and tracking. The typical flow of such a system is illustrated in Figure D.7." [1]

" [8] presents a similar breakdown for traffic sign recognition. The detection and classification stages are executed sequentially on each frame, whereas the tracking stage feeds back spatial and temporal information between frames. The detection problem is concerned with locating TL candidates. Classification is done based on features extracted from the detected candidates. Tracking uses information about location and TL state when tracking TLs through a sequence of frames. A TLR system that addresses the mentioned problems can therefore be broken into 4 stages: detection, feature extraction, classification, and tracking." [1]

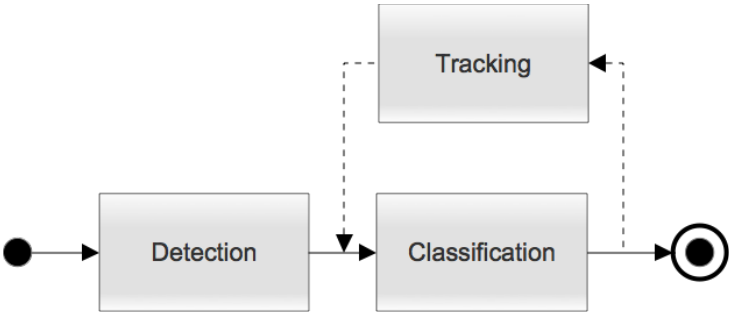


Fig. D.7: *"Breakdown of a computer vision based TLR system."* [1]

4.1 Challenges in recognizing traffic lights for DAS

"When several TLs are simultaneously visible to the driver, each possibly in different light states, the assistance system must be able to determine which TL is relevant to the driver, a task that can be difficult, even to a human. Figure D.8 shows an example of a complex traffic scene where three upcoming intersections are all visible at the same time. One of the intersections contains turn lanes that are accompanied by their own independent TLs. This represents a major challenge for TLR systems in relation to DAS, in determining whether a TL is relevant to the ego-vehicle." [1]

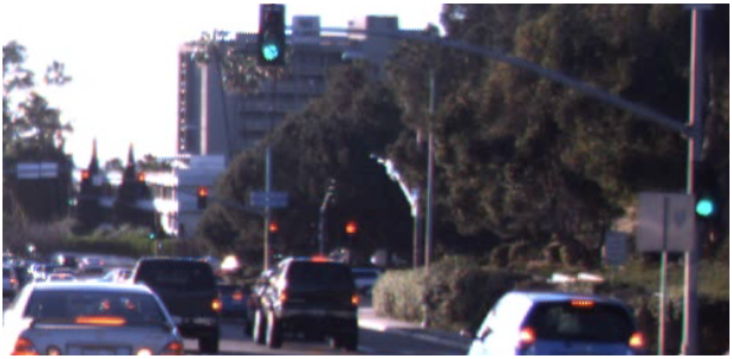


Fig. D.8: *"Complex traffic scene with multiple visible intersections and turn lanes, each with their associated TLs."* [1]

"The relevance of a TL is closely connected to it's placement in relation to the ego-vehicle. Information about the location and direction of the ego-vehicle must therefore be matched with the locations of the TLs. The most advanced system for solving this problem is seen in [36], where a guess is made based on the intersection width and the estimated orientation of the TLs. An alternative and less dynamic approach is used in [33], where the route is recorded beforehand and relevant TLs are manually annotated offline. Features are extracted in the annotated regions, and the system is

then able to recognize the relevant TLs on that specific route." [1]

"A TLR system for DAS must communicate the gathered information to the driver, preferably in a way that is non-intrusive and adds as little as possible to the cognitive load of the driver. Information about the driver's attention can be used to activate a given safety system in case the driver is inattentive or to determine whether a driver has noticed a specific object and should be made aware of it. Hence, fusion of data from looking-in and looking-out sensors can be used [39]. In [40] a large set of looking-in activities: head pose estimation, hand and foot tracking; and looking-out activities: vehicle parameters, lane and road geometry analysis, and surround vehicle trajectories, are fused together to predict driver behavior. The presentation aspect of DAS is outside the scope of this paper." [1]

5 TRAFFIC LIGHT RECOGNITION: STATE-OF-THE-ART

"In this section, we present an overview of methods used in each stage of the pipeline for existing TLR systems. The pipeline is divided into four stages; detection, feature extraction, classification, and tracking. This breakdown was described in chapter 4. In addition to the breakdown in the four stages, papers are also presented with their applied color spaces. This is done since the choice of color space is central to TLR and emphasized in some work e.g. [38]. Table D.1 contains an overview of the applied methods for all the papers from academic institutions. Table D.2 shows a similar overview for industry papers. It should be noted that some of the papers presents more than one approach, whereof only the best performing is listed. Since some of the papers focus on only parts of the problem, and in a few cases it is not apparent what methods were used, some fields are left empty. The paper overview covers papers from 2009-2015, with a single exception of an important paper [41] from 2004, which forms the basis for the more recent paper [33]." [1]

5.1 Color space

"As color is a major characteristic of TLs, it is used extensively in most TLR research. It is primarily used for finding region of interest (ROI) and classifying TL states. The variety in color spaces is large, but the RGB color space is often discussed, as it usually is the color space in which input images are represented. Because color and intensity information are mixed in all the channels of the RGB color space, the information is usually translated to another color space for processing. [37, 42] are the only studies, where RGB is the primary color space. The same author group also utilizes the YCbCr color space in their earliest paper [43]. [44] uses both RGB and YCbCr, in two separate stages, RGB is used for localizing the box of the TL, whereas YCbCr is used for detecting the arrow TL." [1] In [45], CIELab is used when

extracting features for TL detection, subsequent they also employ RGB for extracting features for TL classification. *"Normalized RGB has seen used by it self, as in [46] and combined with RGB in [38, 47, 48]. [34, 35, 49, 50] use grayscale for initial segmentation in the form of spot light detection. Whereas [34, 35, 50], rely purely on grayscale, hence, their systems must function using only intensity and shape information. Other works that use grayscale, are [51], where normalized grayscale is used in addition to CIELab and HSV and [52] where grayscale is used for finding candidates, before determining their state using the HSV color space. The HSV and HSI color spaces are well represented by their use in [49, 52–56], and [57, 58], respectively."* [1] It is noteworthy that [54], demonstrates that the hue distribution is much narrower if a low exposure is used when capturing frames. The narrower distribution greatly helps in later segmentation of the frames by limiting color saturation. For each low exposure frame they also capture one with normal exposure to maintain a balanced intensity. [59] uses the HLS color space for determining the state of found TLs and *" [32] uses IHLS which is an modification of HLS, which separates chromatic and achromatic objects well"* [1] . [60, 61] uses the LUV color space for extracting color features.

"There is no clear tendency towards the use of one particular color space, but color spaces where color and intensity information is separate are clearly preferred. In some recent work such as [38, 51] and to some degree [62], researches have begun combining channels from multiple color spaces in order to get optimal separation of TL colors. In [63, 64], the CieLab color space is used to create a new channel by multiplying the lightness with the sum of the a and b channels." [1]

5.2 Detection

"TLR systems usually look for a selection of TL components. Figure D.9 shows an illustration of the various TL components. The structural relationship between the components is in some cases also used." [1]

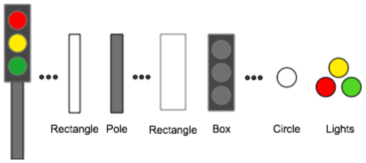


Fig. D.9: Supported TL along with it's different components. [1]

Detection approaches can be categorized as either learning-based or model-based, the latter is currently the far most widely used. It relies on heuristically determined models using shape, color and intensity.

Model-based *"A simple and widely used color model consists of color density thresholds for each TL lamp color, this is seen in [32, 36, 37, 42, 43, 46, 48, 49,* *54, 56, 58, 64]. Detectors based on this kind of simple color model are especially in danger of suffering from overfitting to the specific training set. [33, 41] present several detectors, the best performing being a Gaussian pixel classifier, created from thousands of manually annotated images. In [38, 47], fuzzy clustering is introduced to generate unique representations of a given color. Opposite to regular clustering, sometimes called hard clustering, data points in fuzzy clustering can belong to more than one cluster; the association degree to clusters can vary, meaning that a data point can have a higher probability of belonging to one cluster than another [65]. Common for all detectors relying on color distributions is their sensitivity to disturbances in color which can be caused by many of the effects described subsection 3.1 [38, 47, 48, 53, 54]. Spotlight detection using the white top hat operation is a popular detection approach which is robust to disturbances in color. It is usually based on greyscale images, as seen in [34, 35, 49, 50]. In [53], the V channel from the HSV color space is used with the same effect."* [1]

"Shape models are either used an alternative to the dominating color models or as a filtering step after the color based segmentation In [42] the Hough transform is used on an edge map from a Laplacian edge detection filter. The contribution of [46] is a modified version of the circular Hough transform, which looks for filled circles, and outperforms the conventional circular Hough transform. In [62] they improve this idea further by also looking for other circles around active lights. [32] first apply a Laplacian filter that extracts a clear boundary, disregarding halo effects, before looking for approximate ellipses in the canny edge pixels around candidate BLOBs. In [63, 64], fast radial symmetry is used for finding circles, followed by local maximum and minimum for finding the exact parameters of the given circle. [52] finds object boundaries using morphological operations and thresholding, the borders are topologically analyzed and TL candidate rectangles are located." [1]

"Most of the reviewed work applies BLOB analysis in various degrees for noise removal or for calculating BLOB properties. [34, 35, 37, 43, 44, 49, 50, 57, 58] removes noise by looking at a selection of BLOB features, from relative position of elements such as circles, squares, rectangles, spots, containers, to size, aspect ratio, shape, etc.. An example is [49] where BLOB size, aspect ratio, circular shape, holed regions, and the estimated height in world coordinates is used. [58] employs region growing using seed points from their found BLOBs to perform a border property check between found BLOBs and their corresponding grown regions. Other BLOB analysis includes doing bounding box analysis as in [38, 47, 48, 53, 54], where the goal is to locate the TL box such that the state within it can be estimated. Instead of finding shape from segmented BLOBs, [46, 52] applies a Sobel Kernel to get an edge map and applies Hough transform in order to find either circular shapes or boxes." [1]
Using BLOB analysis to filter out false TL candidates is very dependent on the quality of the BLOB segmentation. In many cases BLOBs from actual TLs will appear vastly different from a TL BLOB under ideal conditions.

Learning-based *"An early attempt at learning based detection was seen in [33, 41] where a cascading classifier based on Haar features was tested. It was, however, outperformed by their Gaussian color classifier. Recently, three papers that employ some more successful learning-based detectors have been published."* [1]. [59] is combining occurrence priors from a probabilistic prior map and detection scores based on SVM classification of Histogram of Oriented Gradients (HoG) features to detect TLs. [45] detects TL pixels by extracting features from color, texture, pixel location and normalized voting histograms and classify them using a JointBoost classifier. In [60, 61] features are extracted as summed blocks of pixels in 10 channels created from transformations of the original RGB frames. The extracted features are classified using depth 2 learning trees in [60]. In [61] detector performance is improved by increasing the learning tree depth to 4 and extending the scale space. Common for the learning-based detectors is their requirement for lots of data and computation. Their advantages are a higher robustness to variation and less tendency to overfitting because of the substantial amount of data used in the training process.

Auxiliary detection *"In [42] GPS position is used to activate the detection system when approaching preannotated intersections. [33, 36, 41, 53, 55, 59] are taking this further by improving detection by including maps containing much more precise prior knowledge of TL locations. The maps are created using accurate GPS measurements and manual annotation of TL position. In [55] they store hue and saturation histograms of each TL during the initial TL mapping. This helps with handling differences in the light emitting properties of individual TLs. In [36] the possible states of the individual TLs is also annotated to further reduce false positives. Relying on prior maps can be a big help to visual TL detection. The maps increase the certainty in TL candidates from the detector which makes it easier to reject false candidates. In cases where a TL has not been mapped e.g. due to roadwork, a high reliance on maps might lead to critical missed detections."* [1]

"Generally, the first step of model-based detection is segmentation of ROI by using either, clustering, distributions, or thresholds. This is followed by either looking for circular objects using Hough transform or fast radial symmetry, or BLOB analysis to filter TL candidates. With learning-based detectors all of this is achieved by the classification of numerous features. Prior knowledge of the route, geographical and temporal information can drastically reduce the ROI, hence reduce the computational requirements and the number of bad candidates." [1]

5.3 Feature Extraction

"Color is a widely used feature for classification. This is seen in [33, 36, 38, 41, 46–48, 51–53, 57, 62], where color densities from segmented areas are used. In [49, 55] the features are based on HSV histograms. Besides color, shape and structure are widely used characteristics from TLs. Shape information includes a wide variety of

features, such as aspect ratio, size, and area. Structural information is the relative positioning of the components of TLs. A TL lamp and the surrounding container is, in many cases, easily distinguishable from the background, making shape, and structural information popular features. [58] uses color template matching. Shape information is combined with structural information in [34, 35, 50], and also color features as seen in [38, 46–48, 51, 62]. In [42, 53], color, shape, and structural information is used as features. In [37], a mix of BLOB width, height, center coordinate, area, extent, sum of pixels, brightness moment, and geometric moment is used as features for their SVM classifier. More advanced feature descriptors are seen in [54], where edge information in the form of HoG, are used as features for image regions containing TL containers. [44] uses 2D Gabor Wavelet features and [43, 56] uses Haar features. [45] extract 21 geometric and color features from TL candidate regions. [32] relies on spatial texture layout for classification, specifically they calculate a Local Binary Pattern (LBP) histogram for the TL as well as for five equally sized regions in each color channel, before creating a feature vector consisting of the concatenated LBP histograms." [1]

Systems relying either color, shape, or structural features will be challenged in varying conditions of the real world. By using multiple types of features containing different types information increase robustness.

5.4 Classification

"For classification of TL candidates [53] utilizes a fusion between scores from structure, shape, color, and geolocation information, which help determine whether a TL should exist at that location. [52] simply estimate the state to be the winner of a majority count on the number of pixels within empirically determined thresholds. [57] decides on a TL state for the entire segmented frame based on a contradiction scheme that selects the optimal light from TL position and size." [1]. [59] classifies TL candidates by subdividing them vertically in three using the color distribution. [44] focus on classifying the arrow type of their TL candidates, this is done by nearest neighbor classification of Gabor image features which are reduced by 2D independent component analysis. [54] classifies the TLs by using HoG features from the TL container and SVM. [1] *"In [33, 41], a neural network is used for determining the state of the found TLs. [58] applies template matching by normalized cross correlation. [34, 35, 50] use adaptive template matching. [49] uses SVM for classification based on HSV histograms. [43, 56] use cascading classifiers based on Haar features. [35] compares their proposed adaptive template matching system with the learning-based AdaBoost cascade Haar feature classifier. Their model based approach proved to substantially outperform their learning based approach. [45] classify their 21 element feature vector using a JointBoost classifier. [32] applies SVM to classify LBP feature vectors in order to determine the state of a TL from it's spatial texture layout."* [1]

Successful classification rely heavily on the quality of the features. *"The* *majority of papers apply a classifier to the extracted features and find the best match by comparison with a selection of trained TL states"* [1]. The remaining papers classify TLs based on heuristics. Classification based on e.g. heuristically determined thresholds, is vulnerable to many of the variations found under real world use. The machine learning-based approaches train a model based on training samples, which requires large amount of data with large variation in order to obtain robustness.

5.5 Tracking

"Tracking is commonly used for noise reduction by suppressing false positives and handling dropouts due to e.g. occlusion. It is evident in Table D.1 and D.2 that approximately half of the presented approaches apply some form of tracking." [1]

"Temporal tracking is used to examine previous frames and determine whether a candidate has been found in the same area earlier and whether it has the same properties as a given candidate in the current frame. This is a simple and straightforward approach used in [47, 48]. [57] reduced false detections by a third by using a temporal decision scheme that makes a final classification based on temporal consistency." [1] A similar approach is seen in [45], where a TL has to be detected in three consecutive frames before it is accepted. It is evident their results that including this type of tracking led to an increase of 12.16% in overall precision, while costing 6.27% in overall recall. *" [49] employs multiple target temporal tracking using predictions of the location of TL from the speed of the ego vehicle. This allows for validation of state changes and smoothed recognition confidence. Additionally, they modify top hat kernel size, saturation, and intensity thresholds when TLs are about to disappear from the field of view. This enables recognition in a greater distance interval. Before reaching the final state verdict, [49] inputs the detected state from their classifier into a range of HMMs, one for each possible type of TL and one for a non-TL objects. The model which best fits the detected sequence of states is then selected as the final estimated state. [52] also employs HMM, although only for a single TL type. [38] estimates the distance to TLs using inverse perspective mapping and tests both a Kalman filter and a particle filter for tracking the relative movement between vehicle and TLs. TLs are then filtered based on their consistency in position and color. In [50], an Interacting Multiple Model filter is used for keeping track of both state and position of a given TL. The prediction in the model is using Kalman filters to keep track of the state and the position in time. For establishing the state, a Markov chain with weighted probabilities is used for finding the current state based on posterior states, originally introduced in [66]. [55] uses prior maps and a histogram filter to adjust the localization mismatch between predict and actual TL area."* [1]

"The correlation tracking used in [33, 41, 53], relies on the fact that a given detected TL's state is unlikely to shift sporadically in a sequence of frames. E.g., when a series of red states are detected, it is most likely that the state in the upcoming frame

will be red again and the appearance will therefore be approximately the same. [56] use CAMSHIFT tracking of candidates across frames based on their appearance." [1]

"Tracking is mostly used to filter out noise and handle lone failed detections, caused by e.g. occlusion. In most of the surveyed papers, tracking consist of a simple temporal consistency check, a few use tracking in a more advanced manner by incorporating prior probabilities." [1] Generally, two types of tracking are used, correlation tracking and point tracking. In many cases correlation tracking rely on the same types of features as the detector and for this reason will be unable to complement the detector when it fails. Point tracking on the other hand can employ temporal information which has a better basis for complementing the detector.

Table D.1: *"Recent academic studies in TLR. Colors indicate paper group affiliation. Corresponding evaluation results and datasets for each paper are available in Table D.3. Abbreviations: Connected component analysis (CCA), support vector machine (SVM), hidden Markov model (HMM)"* [1]

Paper	Year	Color Space(s)	Detection	Features	Classification	Tracking
[38]	2014	RGB, RGB-N	Fuzzy clustering, CCA, BLOB analysis	Color	Color	Kalman filter, particle filter
[47]	2013	RGB, RGB-N	Fuzzy clustering, CCA, BLOB analysis	Color	Color	Temporal filtering
[48]	2012	RGB, RGB-N	Clustering, BLOB analysis	Color	Color	Temporal filtering
[50]	2014	Grayscale	Top-hat spot light detection, BLOB analysis	Shape, structure	Adaptive template matching	Interacting Multiple Model
[35]	2009	Grayscale	Top-hat spot light detection, BLOB analysis	Shape, structure	Adaptive template matching	-
[34]	2009	Grayscale	Top-hat spot light detection, BLOB analysis	Shape, structure	Adaptive template matching	-
[37]	2013	RGB	Color thresholding, BLOB shape filtering	Brightness and geometric moments	SVM	-
[43]	2011	YCbCr	Color thresholding, BLOB shape filtering(width to height ratio, sum of pixels, BLOB area to bounding rectangle ratio)	Haar-like features	-	Adaptive multiclass classifier trained using AdaBoost
[62]	2010	RGB, RGB-N	Pixel clustering, edge map, filled circle Hough transform in neighborhood	Color, shape	Color of best circle	-
[46]	2009	RGB-N	Color thresholding, edge map, filled circle Hough transform	Color, shape	Color of best circle	-
[60]	2015	LUV	Aggregated channel features	-	-	-
[61]	2015	LUV	Aggregated channel features	-	-	-
[45]	2015	RGB, CIELab	Color, texture, pixel location and normalized voting histogram classified with JointBoost	Color, Geometric features	JointBoost	Temporal filtering
[59]	2015	HSL	Probabilistic prior maps and dense HoG	Color	Color distribution	-

Table D.1: *"Recent academic studies in TLR. Colors indicate paper group affiliation. Corresponding evaluation results and datasets for each paper are available in Table D.3. Abbreviations: Connected component analysis (CCA), support vector machine (SVM), hidden Markov model (HMM)"* [1]

Paper	Year	Color Space(s)	Detection	Features	Classification	Tracking
[52]	2014	Grayscale, HSV	Topological analysis of edges	Color	Majority pixel count	HMM
[53]	2014	HSV	Top-hat spot light detection, BLOB analysis	Color, shape, structure	Fusion of color, shape, structure scores, and geolocation	Correlation tracking
[54]	2014	HSV	Color thresholding, BLOB analysis	HoG	SVM	-
[51]	2014	Norm. grayscale, CIELab, HSV	Prior knowledge of TL location	Color, shape	Convolutional neural network	-
[64]	2014	CIELab	Color thresholding, radial symmetry, local maximum and minimum, shape filtering	-	-	-
[44]	2012	RGB, YCbCr	BLOB analysis	2D Gabor wavelet	Nearest neighbor	-
[63]	2012	CIELab	Color difference enhancement, neighborhood image filling, radial Symmetry	Color	Color	Spatial-temporal consistency check
[42]	2012	RGB	Color thresholding, edge map with Laplace filter, circle Hough transform, shape filtering	Color, shape, structure	Color	-
[58]	2011	HSI	Color thresholding, dimensionality and border property check	Color	Normalized cross correlation template matching	-
[32]	2011	IHLS	Color thresholding, Laplacian filter for boundary extraction, approximate ellipses based on edge pixels from canny	LBP features	SVM	-
[55]	2011	HSV	Prior knowledge of traffic light location	HS histograms	Histogram backprojection	Histogram filter
[49]	2010	HSV	Top-hat spot light detection, Color thresholding, CCA, BLOB analysis	Concatenated HSV histogram	SVM	HMM and temporal tracking using ego motion

Table D.1: *"Recent academic studies in TLR. Colors indicate paper group affiliation. Corresponding evaluation results and datasets for each paper are available in Table D.3. Abbreviations: Connected component analysis (CCA), support vector machine (SVM), hidden Markov model (HMM)"* [1]

Paper	Year	Color Space(s)	Detection	Features	Classification	Tracking
[56]	2010	HSV	Color thresholding, morphological operation	Haar features	AdaBoost trained classifier	CAMSHIFT
[57]	2009	HSI	Gaussian-distributed classifier, BLOB analysis, temporal information	Color	Global contradiction solving scheme	Temporal filtering/decision scheme

Table D.2: *"Recent studies in TLR from industry. Colors indicate paper group affiliation. Corresponding evaluation results and datasets for each paper are available in Table D.2."* [1]

Paper	Year	Color Space(s)	Detection	Features	Classification	Tracking
[33]	2013	-	Prior knowledge of TL location, Gaussian-distribution classifier	Color	Neural network	Correlation tracking
[41]	2004	-	Prior knowledge of TL location, Gaussian-distribution classifier	Color	Neural network	Correlation tracking
[36]	2011	-	Prior knowledge of TL location and state sequence	Color, shape	Color and BLOB geometry	Temporal filtering

6 EVALUATION

Performance of TLR systems has been evaluated in a wealth of ways throughout the reviewed work, complicating comparisons between competing approaches. Additionally, some papers does not clearly define which evaluation criteria have been used. Evaluation is generally done on a local collection of frames, unavailable to the public. These local datasets are mostly small in size and contain little variation.

6.1 Performance measures

The most common measures of system performance are: precision, recall, and accuracy. Results from the reviewed TLR systems are therefore, when possible, summarized using these measures. Precision, recall, and accuracy are defined as in [67]. The definitions are shown in equation (D.1), (D.2) and (D.3). TP, FP, FN, TN are abbreviations for true positives, false positives, false negatives, and true negatives. [1]

$$Precision = \frac{TP}{TP + FP} \tag{D.1}$$

"A Precision close to one indicates that all the recognized TL states are in fact correctly recognized." [1]

$$Recall = \frac{TP}{TP + FN} \tag{D.2}$$

"A recall close to one indicates that all the TL states, in a given video sequence, were correctly recognized by the proposed system." [1]

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{D.3}$$

"An accuracy close to one indicates that the system detects all TLs with no false positives. Traditionally, true negatives are also included in the calculated accuracy as follows from equation (D.3)*, but true negatives are rarely used in evaluation of TLR systems."* [1]

"In some cases these performance measures are referred to under different names, e.g. detection rate instead of recall or recognition rate for accuracy. When it is unclear what the used terms are covering, they are published as accuracy in Tables D.3, D.4 and D.5. The criteria for deciding when a TL is recognized and registered as a true positive can also be unclear or vary widely. An example of this is seen in [34, 35], where a TL is registered as a TP if it has been recognized once in the whole series of frames where it appears. We suggest evaluating FPs, TPs on a frame by frame basis, as this gives a more complete representation of a given system's performance." [1]

6.2 Evaluation overview

In Table D.3 evaluation data specifications are presented along with the stated results from TLR papers originating from academic institutions. Table D.4 presents the same for TLR research originating from industry. Since a few papers have published results limited to TL detection, these results are presented separately in Table D.5. When looking at the tables it is clear that the majority of systems are evaluated on local datasets. Many of these datasets are not described sufficiently and consist of as little as 35 ground truth TLs. Taking dataset size and evaluation results in to account, [38] is one of the best performing systems. TLs are detected based on fuzzy clustering, the system has been refined from earlier publications [47, 48] and recognition is supported by adaptive image acquisition and tracking. Another notable system, with impressive results, is presented in [59], where TLs are detected and recognized in an extensive dataset using a probabilistic prior map and detection based on HoG features. *" [33] presents an autonomous car with TLR which has successfully driven a 100 km route in real traffic. TLs are detect using prior maps and from TL color distributions. The performance of the TLR system is, however, not quantified, atleast not publicly, making it impossible to do direct comparisons to other work."* [1]

Table D.3: *"Evaluation datasets and corresponding results from recent studies in TLR from academia. Colors indicate paper group affiliation. RCMP abbreviates Robotics Centre of Mines ParisTech and signifies the use of their dataset, - indicates evaluation on part of the dataset and + indicates the addition of private datasets. Under Ground Truth, # of frames indicates # of frames with a minimum of 1 TL."* [1]

Paper	Year	Dataset	Size [Frames]	Ground Truth	Resolution [Pixels]	Conditions	Pr [%]	Re [%]	Ac [%]
[38]	2014	Local	75,258	19,083 frames	752x480	Day, night	99.38	98.24	99.39
[47]	2013	Local	16,176	4,600 frames	752x480	Night	98.04	-	-
[48]	2012	Local	27,000	14,000 frames	752x480	Day, night	90.32	-	-
[50]	2014	Local	-	-	-	-	97.6	87.57	97.6
[35]	2009	RCMP+	>11,179	10,339 TLs	640x480	Day	84.5	53.5	-
[34]	2009	RCMP+	>11,179	>9,168 TLs	640x480	Day	95.38	98.41	-
[37]	2013	Local	16,080	12,703 frames	620x480	Night	-	93.53	-
[43]	2011	Local	30,540	16,561 frames	620x480	-	-	93.80	-
[62]	2010	Local	35	35 TLs	-	-	-	-	89.0
[46]	2009	Local	30	30 TLs	-	-	-	-	86.67
[45]	2015	Local, RCMP-	-	-	640x480+	Day, night	72.83	80.13	-
[59]	2015	Local	-	9,301 frames	6x1024x768	Early morning, afternoon	92.3	99.0	-
[52]	2014	Local	649	446 TLs	648x488	Mixed day-time illumination conditions	99.59	92.19	94.45
[53]	2014	Local	3,767	-	-	Day	-	-	96.07

Table D.3: *"Evaluation datasets and corresponding results from recent studies in TLR from academia. Colors indicate paper group affiliation. RCMP abbreviates Robotics Centre of Mines ParisTech and signifies the use of their dataset, - indicates evaluation on part of the dataset and + indicates the addition of private datasets. Under Ground Truth, # of frames indicates # of frames with a minimum of 1 TL."* [1]

Paper	Year	Dataset	Size [Frames]	Ground Truth	Resolution [Pixels]	Conditions	Pr [%]	Re [%]	Ac [%]
[54]	2014	Local	-	-	640x480	Mixed day-time illumination conditions	-	-	-
[51]	2014	Local	3,351	3,351 TLs	-	Afternoon, dusk	-	-	97.83
[44]	2012	Local	5,000	-	1392x1040	Mixed day-time illumination conditions	-	-	91.00
[63]	2012	RCMP	11,179	9,168 TLs	640x480	Day	61.22	93.75	-
[42]	2012	Local	7,311	-	-	Day	-	89.9	-
[58]	2011	RCMP-	5,553	5,553 frames	640x480	-	96.95	94.4	-
[32]	2011	Local	714	763 TLs	640x480	Sunny, cloudy, rainy	34.51	94.63	95.01
[55]	2011	Local	-	-	1280x1024	Noon, dusk, night	81.46	77.98	92.85
[49]	2010	Local	-	2,867 TLs	512x384	-	-	-	89.6
[56]	2010	Local	-	-	780x580	-	-	-	-
[57]	2009	Local	6,630	-	640x480	-	-	-	98.81

Table D.4: Evaluation datasets and corresponding results from recent studies in TLR originating from industry. Colors indicate paper group affiliation. Under *Ground Truth*, # of frames indicates # of frames with a minimum of 1 TL.

Paper	Year	Dataset	Size [Frames]	Ground Truth	Resolution [Pixels]	Conditions	Pr [%]	Re [%]	Ac [%]
[33]	2013	Local	-	-	-	100 km in real world	-	-	-
[41]	2004	Local	-	-	-	-	-	>95	-
[36]	2011	Local	-	1,383 frames	2040x1080	Morning, afternoon, night	99.65	61.94	93.63

Table D.5: *"Evaluation datasets and results from TL detection papers. Under Ground Truth, # of frames indicates # of frames with a minimum of 1 TL."* [1]

Paper	Year	Dataset	Size [Frames]	Ground Truth	Resolution [Pixels]	Conditions	Pr [%]	Re [%]	Ac [%]
[60]	2015	LISA TL	14,386	21,421 TLs	1280x580	Mixed day-time illumination conditions	30.1	50.0	-
[61]	2015	LISA TL	11,527	42,718 TLs	1280x580	Night-time	65.2	50.0	-
[59]	2015	Local	-	9,301 frames	6x1024x768	Early morning, afternoon	97.3	99.0	-
[64]	2014	Local	70	142 TLs	240x320	Day	84.93	87.32	-

6.3 Proposed evaluation methodology

"A variety of performance measures can be used for evaluation of TLR systems, examples are recognition rate, detection rate, recall, precision, true positive rate, false positive rate, false positives per frame, confusion matrix, F1-score, etc. No matter which measure is used, it is important to define it clearly. Furthermore, it is important to describe the used datasets in detail in order to make fair assessments possible." [1]

True positive criteria

"It should be clear what constitute a true positive. We suggest using the PASCAL overlap criterion introduced in [68]. It is defined as seen in equation (D.4).*"* [1]

$$a_0 = \frac{\text{area}(B_d \cap B_{gt})}{\text{area}(B_d \cup B_{gt})} \geq 0.5 \tag{D.4}$$

"a_0 denotes the overlap ratio between the detected bounding box B_d and the ground truth bounding box B_{gt}." [1]

Describing performance

"According to [69] it can be misleading to state performance using overall accuracy on an unevenly distributed dataset. The overall accuracy does not necessarily represent the performance on smaller classes." [1] The distribution of classes in TL datasets is naturally skewed since e.g. the standard stop light is a lot more common than the warning arrow light. When evaluating multi-class classifiers on skewed datasets, confusion matrices provide a great overview of the performance for specific classes. *"In table D.6 we present an example of a confusion matrix for a basic TLR system with the stop, warning, and go classes. Dark grey indicates the variables, and brighter grey indicate either ground truth or system classification output. From these numbers, recall, precision, and accuracy seen*

in the blue fields, can be calculated. The classifications in the confusion matrix for this example were found based on thresholds that provide a high recall at the cost of precision." [1]

Table D.6: Confusion matrix of 3-class skewed dataset. [1]

			System classification			
			Stop	Warning	Go	Recall
			9210	5102	6194	
Ground Truth	Stop	9703	5.887	160	0	60.67%
	Warning	691	1	406	0	58.76%
	Go	7688	2	0	4.872	63.37%
	Precision		63.92%	7.96%	78.66%	40.71%

The vast majority of papers report performance using the three measures described in subsection 6.1. The problem with these measures is that they only provided a narrow glimpse into the actual performance of the system. By calculating accuracy or precision and recall for a large number of thresholds and plotting the resulting curves, it is possible to observe the performance over the full spectrum of the system. *"The two most common types of curves are Receiver Operator Characteristic (ROC) curves, cost curve, and Precision-Recall (PR) curves. In [70], the relationship between ROC curves and PR curves is presented, and it is concluded that when using skewed datasets, the PR curves provides a more informative picture of a system's performance. Both [70] and [69] mention Area-Under-Curve (AUC) as an alternative to the traditional measures for comparing algorithms. AUC can describe performance in the full spectrum using a single number. When calculating the AUC it is important to keep in mind that using few thresholds as basis for generating the curve may lead to a poor representation of the systems performance. Ideally the number of thresholds should match the number of different scores outputted by the TLR system."* [1]

Figure D.10 shows PR curves for the same TLR example used in table D.6. The optimal threshold for a given system can be easily determined using the PR curves. Additionally, the AUC will reflect the dramatic drop in precision of the stoplight classifier, a single precision, or accuracy measure on the other hand could not.

"The proposed evaluation terms and criteria are listed below:" [1]

- *"True positives are defined according to the PASCAL overlap criterion."* [1]
- *"Precision, as seen in equation* (D.1)*."* [1]

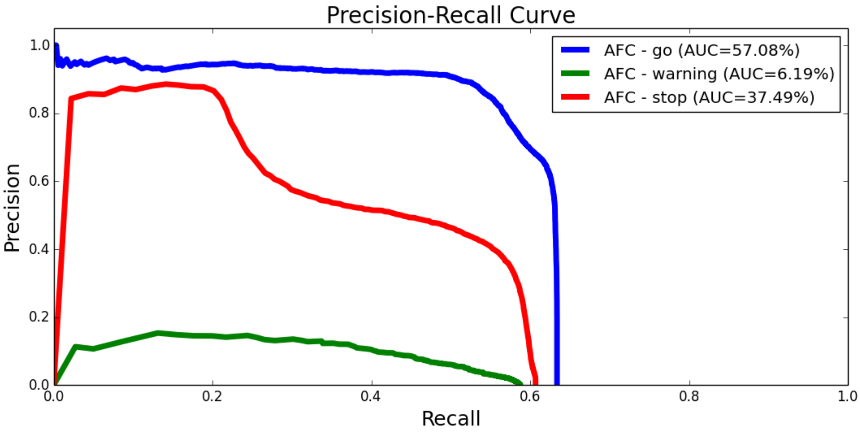


Fig. D.10: Recognition performance on skewed dataset.

- *"Recall, as seen in equation* (D.2)*."* [1]
- *"Area-Under-Curve for Precision-Recall curve."* [1]
- *"Confusion matrix."* [1]

7 TRAFFIC LIGHT DATASET

Extensive and challenging datasets are essential for evaluation and comparison of TLR research. Until now the only publicly available dataset was the TLR benchmark from *LaRA (La Route Automatisée)* at Mines ParisTech, Paris. In this section the new Traffic Light Dataset from *LISA (Laboratory for Intelligent and Safe Automobiles)* at University of California, San Diego, is described in detail. Table D.7 provides an overview of these two TL datasets.

"The LISA Traffic Light Dataset consists of TLs that are found in San Diego, California, USA. The dataset provides two day and two nighttime sequences for testing. These test sequences contain 23 minutes and 25 seconds of driving around San Diego. The stereo image pairs are acquired using the Point Grey's Bumblebee XB3 (BBX3-13S2C-60) which is constructed with three lenses which each capture images with a resolution of 1280x960. The lenses have a Field of View(FoV) of 66°. Because of the 3 lenses, the stereo camera supports two different baselines, 12 and 24 cm, whereof the widest is used for the LISA Traffic Light Dataset. The stereo images are uncompressed and was rectified on the fly. The Bumblebee XB3 was mounted in the center of the roof of the capturing vehicle and connected to a laptop by FireWire-800 (IEEE-1394b). Besides the 4 test sequences, 18 shorter video clips are provided for training and testing. Gain and shutter speed were manually set to avoid over saturation as well as limit the effect of flickering from the TLs. For all day clips,

Table D.7: *"Overview of existing public TL databases. The ambiguous class covers uncertain annotations."* [1]

	LaRA, Mines ParisTech [71]	*LISA, UCSD*
#Classes	4 (green, orange, red, & ambiguous)	7 (go, go forward, go left, warning, warning left, stop, & stop left)
#Frames/#GT	11,179 / 9,168	43,007 / 119,231
Frame spec.	Mono, 640 x 480, 8-bit, RGB	Stereo, 1280 x 960, 8-bit, RGB
Video	Yes, 8min 49s @ 25FPS	Yes, 44min 41s @ 16FPS
Description	1 sequence, urban, day, Paris, France	4 test seq. $\geq$ 4min and 18 training clips $\leq$ 2min 49s, urban, morning, evening, night, San Diego, USA

shutter speed was 1/5000 sec and gain was set to 0. For all night clips, shutter speed was 1/100 sec and gain was set to 8. A Triclops calibration file is provided along with the stereo images, this file contains the factory calibration for the used Bumblebee XB3 camera. Table D.8 shows a detailed list of the short video clips and longer video sequences that constitute the LISA Traffic Light Dataset." [1]

Table D.8: *"Overview of the video sequences in LISA Traffic Light Dataset."* [1]

Sequence name	**Description**	**# Frames**	**# Annotations**	**# TLs**	**Length**	**Classes**
Day seq. 1	morning, urban, backlight	4,060	10,308	25	4min 14s	Go, warning, warning left, stop, stop left
Day seq. 2	evening, urban	6,894	11,144	35	7min 11s	Go, go forward, go left, warning, stop, stop left
Night seq. 1	night, urban	4,993	18,984	25	5min 12s	Go, go left, warning, stop, stop left
Night seq. 2	night, urban	6,534	23,734	62	6min 48s	Go, go left, warning, stop, stop left
Day clip 1	evening, urban, lens flare	2,161	10,372	10	2min 15s	Go, warning, stop, stop left
Day clip 2	evening, urban	1,031	2,230	6	1min 4s	Go, go left, warning left, stop, stop left
Day clip 3	evening, urban	643	1,327	3	40s	Go, warning, stop
Day clip 4	evening, urban	398	859	8	24s	Go
Day clip 5	morning, urban	2,667	9,717	8	2min 46s	Go, go left, warning, warning left, stop, stop left

Table D.8: *"Overview of the video sequences in LISA Traffic Light Dataset."* [1]

Sequence name	Description	# Frames	# Annotations	# TLs	Length	Classes
Day clip 6	morning, urban	468	1,215	4	29s	Go, stop, stop left
Day clip 7	morning, urban	2,719	8,189	10	2min 49s	Go, go left, warning, warning left, stop, stop left
Day clip 8	morning, urban	1,040	2,025	8	1min 5s	Go, go left, stop, stop left
Day clip 9	morning, urban	960	1,940	4	1min	Go, go left, warning left, stop, stop left
Day clip 10	morning, urban	40	137	4	3s	Go, stop left
Day clip 11	morning, urban	1,053	1,268	6	1min 5s	Go, stop
Day clip 12	morning, urban	152	229	3	9s	Go
Day clip 13	evening, urban	693	1,256	8	43s	Go, warning, stop
Night clip 1	night, urban	591	1,885	8	36s	Go
Night clip 2	night, urban	2,300	4,607	25	2min 23s	Go, go left, warning, warning left, stop, stop left
Night clip 3	night, urban	1,051	2,027	14	1min 5s	Go, go left, warning left, stop, stop left
Night clip 4	night, urban	1,105	2,536	9	1min 9s	Go, warning, stop
Night clip 5	night, urban	1,454	3,242	19	1min 31s	Go, go left, warning, stop, stop left
		43,007	119,231	304	44min 41s	

The LISA Traffic Light Dataset is captured in stereo since stereo vision is widely used in related computer vision areas and might see more use in TLR. *"Both mono and stereo vision are widely used for vehicle detection according to [72], which review vehicle detectors. Additionally, [72] describe a stereo vision bottom-up paradigm which consist of visual odometry, feature points in 3D, and distinguishing static from moving points, which is also mentioned in [73]. All parts of this paradigm can potentially reduce the amount of false positives. The benefit of stereo vision is reinforced by [74] where the main technical challenges in urban environments are said to be occlusions, shadow silhouettes, and dense traffic. The introduction of stereo has shown promising results in relation to solving these challenges."* [1]

"Each sequence in the dataset comes with hand labeled annotations for the left stereo frame. Annotations for a given video sequence contains the following information: frame number, rectangular area around the lit TL lamp, and it's state." [1] A heatmap of the all annotations in the dataset can be seen in Figure D.11, where it is clear that most of the annotations are done in the upper right part

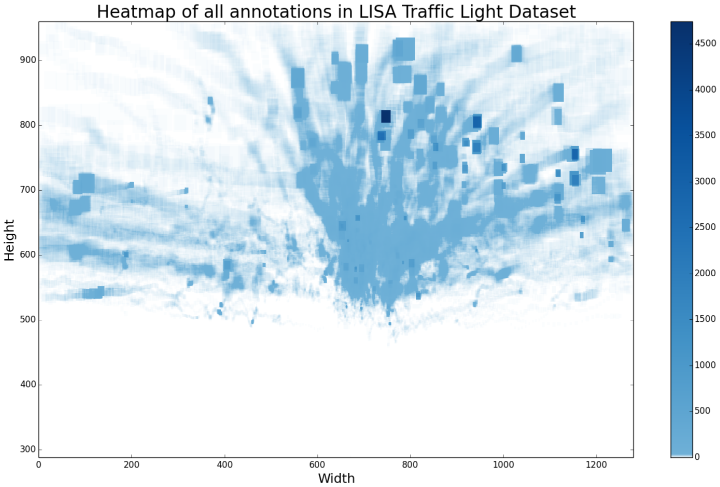


Fig. D.11: Heatmap of all annotations in the LISA TL Dataset.

of the frame, and a few TLs are annotated in the far left side. It is therefore safe to reduce the search for TL to the upper part of the frames.

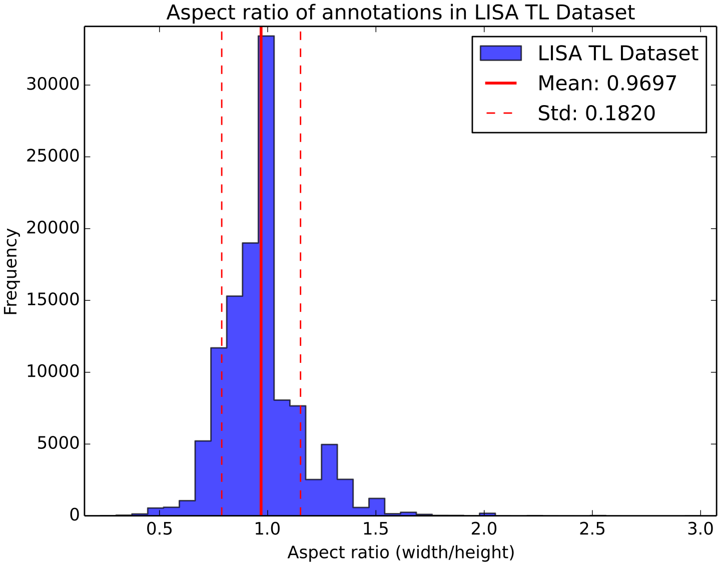


Fig. D.12: Aspect ratio histogram of LISA TL Dataset.

Figure D.12 shows a histogram of the aspect ratio of all the annotations in the dataset. The mean aspect ratio is 0.9697, which fits the quadratic TL bulbs well. The variation in aspect ratio is caused by viewing angles, motion blur and imprecise annotation.

"The LISA Traffic Light Dataset is made freely available at `http://cvrr.ucsd.edu/LISA/datasets.html` *for educational, research, and non-profit purposes."* [1]

8 DISCUSSION AND PERSPECTIVES

"In this section we discuss the current trends and perspectives based on the surveyed papers. It is difficult to determine the state of TLR as evaluation is done on local datasets and with different evaluation methodology. To maintain and advance research on TLR, it is essential to use a common evaluation methodology on challenging publicly available TL datasets. This will enable both newcomers and established research groups to efficiently compare approaches. [71] provides the only publicly available dataset. It is unfortunately not widely used and lacks variation. The ideal TLR benchmark should have large variation in environmental conditions, similar to The KITTI Vision Benchmark Suite for evaluation of stereo correspondence [75], and the VIVA challenge [76] for hands, face, and traffic signs. When TLR systems eventually matures, the evaluation metrics should evolve to include weighted penalties for missed or wrong recognitions based on the severity of the error. Furthermore, the distance where TLR systems are first able to successfully recognize a TL is very relevant and should also be part of the evaluation. Since no comprehensive surveys have existed until now, it required substantial effort to gain an overview of the state of TLR research. The scope of existing TLR research vary significantly, spanning from very basic TL detection, to complex systems robust enough to be used in autonomous systems as seen with [33]. Table D.3 indicates that many of the surveyed systems performs in the high 90% in recall, precision, and accuracy. The best performing papers seem to be [41], [33], [36] from the industry, and [55], [53], [59] from academic institutions. The approaches from these papers rely on prior maps of TL location and properties, which makes it possible to achieve solid performance under challenging conditions. Such systems can reduce the number of false positives substantially, because the approximate locations of TLs are known." [1] Using information from precise maps is a big advantage over conventional systems. [59] shows that their use of prior maps increase precision from 56.67% to 97.33%. *"The price is less flexibility and high cost, since the maps must be kept up to date for the systems to function."* [1]

"The paradigm change from heuristic models to learning-based seen in traffic sign detection and pedestrian detection has not happened for TLR yet." [1] This is underlined by examining Table D.1 where detection of candidate TLs is almost entirely based on heuristic models, with the exception of two very recent papers. In [59] detection is done using HoG and [60] uses the ACF framework. Learning-based detectors have been tried earlier, but do not appear in Table D.1, *"since only the best performing approach from each paper is listed. [35, 41] de-*

veloped learning-based TL detectors, based on Haar features, to compare with their model-based systems." [1] In both cases a model-based detector outperformed the learning-based detector in both detection and computational load.

"The additional information that stereo vision provides is rarely used, one exception to this is [41] where stereo vision is used to measure real world distance and size of detected objects. Doing this resulted in a ten fold decrease in false candidates, along with the tracking benefits of knowing the distance and size. As discussed in [74], stereo vision has proven useful to improve the robustness of computer vision systems. Stereo vision cues could be considered as an additional feature channel or for rejecting false positives from e.g. tail lights and reflections. In [77] vehicle detection at night is assisted by a stereo vision 3D edges extractor, while in [78], vehicle detection rely solely on stereo vision for both day- and nighttime data." [1]

"Less than half of the TLR papers include tracking. The most common use of tracking is a simple temporal consistency check. This efficiently suppress FPs and lone FNs. A few papers uses more advanced and sophisticated tracking, such as HMM, IMM, and CAMSHIFT. This is an area that must be researched further as tracking is known to increase performance as seen in [15] where introduction of tracking to traffic signs recognition significantly reduced the number of FPs. For vehicle detection, [79] has similarly increased performance based on vehicle tracking fused with lane localization and tracking. " [1]

DAS applications for TLR

"There are many applications in which TLR can be used as part of DAS. Table D.9 lists applications which have been mentioned in the surveyed papers." [1]

Table D.9: *"DAS applications for TLR."* [1]

DAS feature	Description	Requirements	References
Dashboard visualization	TL state visualization in dashboard	TLR recognition, lane understanding	[49], [38]
Get going alert	Draw attention to the recently switched light	TLR recognition	[49]
Warn driver of stop light	Drawing attention to upcoming stop light	TLR recognition, intersection, lane understanding	[64], [53], [37]
Stop at stop light	Autonomous vehicle stopping at stop light	TLR recognition, lane understanding	[49]
Smooth stop at stop light	Smooth braking towards stop line at stop light	TLR recognition, stop line, lane understanding	[38]
Stop and go	Automatic stop/start of engine at stop lights	TLR recognition	[49], [38]

"Fusion of data from multiple systems and sensors can greatly improve the overall capabilities of DAS. In [27, 28] the driver's attention is measured using cameras looking inside the car. In [26] a first person view camera is used for capturing. The

driver's registered attention can e.g. be used to activate safety systems in case of the driver being inattentive. By fusing TL recognition with looking-in systems which e.g detect the driver's eye gaze, it can be determined whether or not the driver have noticed the TL. Other properties which can be used to decide if the driver should be informed are the TL's detectability and discriminability as discussed in [80]. Velocity information from the CAN bus can also be obtained to help determining if the vehicle is slowing down while approaching the TL. A lot of applications require fusion of information from multiple systems, this includes most of the application seen in Table D.9. Understanding the traffic scene is necessary as seen with the use of intersection and lane information. A major challenge for TLR in complex intersections is to determine which TLs are relevant to the driver. Selecting the biggest and closest one, as in [81], is a simplistic way of determining which lights to adhere to. In complex intersections, this will not be sufficient and more intelligent approaches must be applied. So far the most intelligent systems for solving this problem is seen in [36], where a guess is made based on the intersection width and the estimated orientation of the TLs. An alternative and less dynamic approach is seen in [33], where relevant TLs are manually annotated before hand. Features are extracted in the annotated regions and the system then recognize relevant TLs on that specific route. " [1]

"TLR can potentially help people by decreasing fatigue and stress level when driving. This is especially true for people with color vision deficiency or similar challenges. As mentioned in the introduction, a large portion of accidents are connected to intersections and red light running. Integration of TLR systems in vehicles can reduce these accidents. Furthermore, the integration of TLR systems and DAS in cars can to some degree be implemented on smartphones as seen with [42, 82]. Another application for a developed TLR system could be naturalistic driving studies (NDS) analysis by automatic detection of events related to e.g. red light running at intersections. Something similar was done with lane detection in [83], where a set of NDS events are identified and quantified." [1]

Directions

"Even though the Daimler group in Germany and the VisLab group in Italy, have successfully managed to make autonomous vehicles drive on public roads, the TLR problem is not considered solved. TLR systems remain challenged by changing weather and light conditions. To overcome these challenges, TLR systems should be able to adapt parameters throughout the TLR pipeline to the changing conditions. Another major problem that still remains to be solved is determining the relevance of recognized TLs. More research should be made into extending TLR for DAS with lane detection, detailed maps, and other traffic scene information." [1] A few learning-based TL detectors have recently been published [59, 60]. It is not possible at this time to tell whether learning-based detectors are superior to heuristic model-based detector. To determine this, more research in applying learning-based detectors for TLR is needed, as well as evaluations on

common datasets.

9 CONCLUDING REMARKS

"This survey presented an overview of the current state of traffic light recognition (TLR) research in relation to driver assistance systems. The approaches from the surveyed paper were broken down into choices made for color space, detection, features, classification, and tracking. For color space, there exist no clear tendency towards one in particular. We have seen a raising popularity for combining channels from multiple color spaces to create a combined color space that separates traffic light (TL) colors well. Most detection approaches rely on color or shape for finding TL candidates other rely on spotlight detection in a single intensity channel. BLOB analysis is generally used to remove bad TL candidates, this is done based on prior knowledge of the properties of TL BLOBs. Furthermore, some of the best performing approaches use detailed maps of the route and temporal information to improve performance. Many papers utilize manually specified models of TLs, which consist of color, shape, and structural features, to do state detection of TL candidates. Other use trained features such as HoG, LBP, and 2D Gabor wavelets, classified using SVM. A few rely on template matching or neural networks using the color and/or shape. The tracking stage is dominated by temporal filtering, while more advanced approaches include HMM, IMM, and CAMSHIFT." [1]

"TLR is dominated by model based approaches, especially for finding TL candidates. This raises the question of whether model based approaches outperform learning based approaches for TLR. Based on the limited experiences with learning based detection this question cannot yet be answered. Additionally, because the systems are evaluated using different methodology and on very different datasets it is not clear which approaches are the best. Only one public dataset with TLs is currently available and it is not widely used. We have therefore contributed a new dataset, the LISA Traffic Light Dataset, which contains TLs captured with a stereo camera in San Diego, USA under varying conditions. The dataset is supposed to enable comparable evaluation on a large and varied dataset, and provides the possibility of including stereo vision for improving TLR. The dataset will be included in the next VIVA Challenge [76]." [1]

REFERENCES

[1] M. P. Philipsen and M. B. Jensen, “Computer Vision at Intersections: Explorations in Driver Assistance Systems and Data Reduction for Naturalistic Driving Studies,” Master’s thesis, Aalborg University, Denmark, June 2015, https://projekter.aau.dk/projekter/files/213565236/main.pdf.

[2] J. Sussman, *Perspectives on Intelligent Transportation Systems (ITS)*. Springer US, 2005.

[3] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, pp. 84–95, 2009.

[4] Federal Highway Administration. (2009) Traffic control devices: Uses and misuses. [Online]. Available: http://safety.fhwa.dot.gov/intersection/resources/fhwasa10005/brief_3.cfm

[5] AAA Foundation for Traffic Safety. (2014) 2014 traffic safety culture index. [Online]. Available: https://www.aaafoundation.org/sites/default/files/2014TSCIreport.pdf

[6] Federal Highway Administration. (2009) Engineering countermeasures to reduce red-light running. [Online]. Available: http://safety.fhwa.dot.gov/intersection/resources/fhwasa09027/resources/intersection%20safety%20issue%20brief%206.pdf

[7] D. Shinar, *Traffic Safety and Human Behavior*, ser. Traffic Safety and Human Behavior. Emerald, 2007, no. vb. 5620.

[8] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1484–1497, 2012.

[9] The Insurance Institute for Highway Safety (IIHS). (2015) Red light running. [Online]. Available: http://www.iihs.org/iihs/topics/t/red-light-running/topicoverview

[10] Vermont Agency of Transportation. (2009) An evaluation of dilemma zone protection practices for signalized intersection control. [Online]. Available: http://vtransplanning.vermont.gov/sites/aot_program_development/files/documents/materialsandresearch/completedprojects/Dilemma_Zone_Final_Report_6_19_09.pdf

[11] M. Diaz, P. Cerri, G. Pirlo, M. Ferrer, and D. Impedovo, "A survey on traffic light detection," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, ser. Lecture Notes in Computer Science, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Springer International Publishing, 2015, vol. 9281, pp. 201–208. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23222-5_25

[12] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Ongoing work on traffic lights: Detection and evaluation," *12th IEEE Advanced Video and Signal-based Survaeillance Conference*, 2015.

[13] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition," in *Proceedings of the 16th Mini-EURO Conference and 10th Meeting of EWGT*, 2005, pp. 644–653.

[14] R. O'Malley, M. Glavin, and E. Jones, "Vehicle detection at night based on tail-light detection," in *1st international symposium on vehicular computing systems, Trinity College Dublin*, 2008.

[15] A. Mogelmose, D. Liu, and M. M. Trivedi, "Traffic sign detection for us roads: Remaining challenges and a case for tracking," in *IEEE Transactions on Intelligent Transportation Systems*, 2014, pp. 1394–1399.

[16] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition - how far are we from the solution?" in *ICJNN*, 2013.

[17] Y.-L. Chen, C.-T. Lin, C.-J. Fan, C.-M. Hsieh, and B.-F. Wu, "Vision-based nighttime vehicle detection and range estimation for driver assistance," in *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 2988–2993.

[18] C. Idler, R. Schweiger, D. Paulus, M. Mahlisch, and W. Ritter, "Realtime vision based multi-target-tracking with particle filters in automotive applications," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 188–193.

[19] S. Gormer, D. Muller, S. Hold, M. Meuter, and A. Kummert, "Vehicle recognition and ttc estimation at night based on spotlight pairing," in *12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–6.

[20] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 453–462, 2010.

[21] J. C. Rubio, J. Serrat, A. M. López, and D. Ponsa, "Multiple-target tracking for intelligent headlights control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 594–605, 2012.

[22] R. O'malley, M. Glavin, and E. Jones, "Vision-based detection and tracking of vehicles to the rear with perspective correction in low-light conditions," *IET Intelligent Transport Systems*, vol. 5, pp. 1–10, 2011.

[23] S. Eum and H. G. Jung, "Enhancing light blob detection for intelligent headlight control using lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 1003–1011, 2013.

[24] R. Satzoda and M. Trivedi, "On enhancing lane estimation using contextual cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[25] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, pp. 1–19, 2012.

[26] A. Tawari, A. Mogelmose, S. Martin, T. B. Moeslund, and M. M. Trivedi, "Attention estimation by simultaneous analysis of viewer and view," in *IEEE 17th International Conference on Intelligent Transportation Systems*, 2014, pp. 1381–1387.

[27] A. Tawari, K. H. Chen, and M. M. Trivedi, "Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation," in *IEEE 17th International Conference on Intelligent Transportation Systems*, 2014, pp. 988–994.

[28] A. Tawari, S. Sivaraman, M. Trivedi, T. Shannon, and M. Tippelhofer, "Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 115–120.

[29] California Department of Transportation. (2015) California manual on uniform traffic control devices. [Online]. Available: http://www.dot.ca.gov/hq/traffops/engineering/control-devices/trafficmanual-current.htm

[30] Federal Highway Administration. (2015) Manual on uniform traffic control devices. [Online]. Available: http://mutcd.fhwa.dot.gov/

[31] United Nations. (2006) Vienna convention on road signs and signals. [Online]. Available: www.unece.org/trans/conventn/signalse.pdf

[32] C.-C. Chiang, M.-C. Ho, H.-S. Liao, A. Pratama, and W.-C. Syu, "Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts," *International Journal of Innovative Computing, Information and Control*, vol. 7, pp. 6919–6934, 2011.

[33] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich, "Making bertha see," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 214–221.

[34] R. de Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 358–363.

[35] R. Charette and F. Nashashibi, “Traffic light recognition using image processing compared to learning processes,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 333–338.

[36] N. Fairfield and C. Urmson, “Traffic light mapping and detection,” in *Proceedings of ICRA 2011*, 2011.

[37] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, “Night-time traffic light detection based on svm with geometric moment features,” *World Academy of Science, Engineering and Technology 76th*, pp. 571–574, 2013.

[38] M. Diaz-Cabrera, P. Cerri, and P. Medici, “Robust real-time traffic light detection and distance estimation using a single camera,” *Expert Systems with Applications*, pp. 3911–3923, 2014.

[39] M. M. Trivedi, T. Gandhi, and J. McCall, “Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 108–120, 2007.

[40] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, “On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems,” *Computer Vision and Image Understanding*, vol. 134, pp. 130 – 140, 2015.

[41] F. Lindner, U. Kressel, and S. Kaelberer, “Robust recognition of traffic signals,” in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 49–53.

[42] E. Koukoumidis, M. Martonosi, and L.-S. Peh, “Leveraging smartphone cameras for collaborative road advisories,” *IEEE Transactions on Mobile Computing*, vol. 11, pp. 707–723, 2012.

[43] H.-K. Kim, J. H. Park, and H.-Y. Jung, “Effective traffic lights recognition method for real time driving assistance system in the daytime,” *World Academy of Science, Engineering and Technology 59th*, 2011.

[44] Z. Cai, Y. Li, and M. Gu, “Real-time recognition system of traffic light in urban environment,” in *IEEE Symposium on Computational Intelligence for Security and Defence Applications (CISDA)*, 2012, pp. 1–6.

[45] V. Haltakov, J. Mayr, C. Unger, and S. Ilic, “Semantic segmentation based traffic light detection at day and at night,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science, J. Gall, P. Gehler, and B. Leibe, Eds. Springer International Publishing, 2015, vol. 9358, pp. 446–457. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24947-6_37

[46] M. Omachi and S. Omachi, "Traffic light detection with color and edge information," in *2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 284–287.

[47] M. Diaz-Cabrera and P. Cerri, "Traffic light recognition during the night based on fuzzy logic clustering," in *Computer Aided Systems Theory-EUROCAST 2013*. Springer Berlin Heidelberg, 2013, pp. 93–100.

[48] M. Diaz-Cabrera, P. Cerri, and J. Sanchez-Medina, "Suspended traffic lights detection and distance estimation using color features," in *15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1315–1320.

[49] D. Nienhuser, M. Drescher, and J. Zollner, "Visual state estimation of traffic lights using hidden markov models," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1705–1710.

[50] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "Tracking both pose and status of a traffic light via an interacting multiple model filter," in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[51] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in *IEEE 17th International Conference on Intelligent Transportation Systems*, 2014, pp. 2286–2291.

[52] A. Gomez, F. Alencar, P. Prado, F. Osorio, and D. Wolf, "Traffic lights detection and state estimation using hidden markov models," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 750–755.

[53] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, "A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles," in *33rd Chinese Control Conference (CCC)*, 2014, pp. 4924–4929.

[54] C. Jang, C. Kim, D. Kim, M. Lee, and M. Sunwoo, "Multiple exposure images based traffic light recognition," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1313–1318.

[55] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 5784–5791.

[56] J. Gong, Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen, "The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 431–435.

[57] Y. Shen, U. Ozguner, K. Redmill, and J. Liu, "A robust video based traffic light detection algorithm for intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 521–526.

[58] C. Wang, T. Jin, M. Yang, and B. Wang, "Robust and real-time traffic lights recognition in complex urban environments," *International Journal of Computational Intelligence Systems*, vol. 4, no. 6, pp. 1383–1390, 2011.

[59] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[60] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," *18th IEEE Intelligent Transportation Systems Conference*, 2015.

[61] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection at night: Comparison of a learning-based detector and three model-based detectors," *11th Symposium on Visual Computing*, 2015.

[62] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 809–812.

[63] G. Siogkas, E. Skodras, and E. Dermatas, "Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information." in *VISAPP (1)*, 2012, pp. 620–627.

[64] S. Sooksatra and T. Kondo, "Red traffic light detection using fast radial symmetry transform," in *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2014, pp. 1–6.

[65] S. Naz, H. Majeed, and H. Irshad, "Image segmentation using fuzzy clustering: A survey," in *Emerging Technologies (ICET), 2010 6th International Conference on*, 2010, pp. 181–186.

[66] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, pp. 780–783, 1988.

[67] D. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer Berlin Heidelberg, 2008. [Online]. Available: https://books.google.dk/books?id=2vb-LZEn8uUC

[68] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[69] C. G. Weng and J. Poon, “A new evaluation measure for imbalanced datasets,” in *The 7th Australasian Data Mining Conference-Volume 87*. Australian Computer Society, Inc., 2008, pp. 27–32.

[70] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *The 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[71] Robotics Centre of Mines ParisTech. (2015) Traffic lights recognition (tlr) public benchmarks. [Online]. Available: http://www.lara.prd.fr/benchmarks/trafficlightsrecognition

[72] S. Sivaraman and M. M. Trivedi, “A review of recent developments in vision-based vehicle detection.” in *Intelligent Vehicles Symposium*. IEEE, 2013, pp. 310–315.

[73] R. Danescu, F. Oniga, and S. Nedevschi, “Modeling and tracking the driving environment with a particle-based occupancy grid,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 1331–1342, 2011.

[74] N. Buch, S. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 920–939, 2011.

[75] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[76] U. Laboratory for Intelligent and Safe Automobiles . (2015) Vision for intelligent vehicles and applications (viva) challenge. [Online]. Available: http://cvrr.ucsd.edu/vivachallenge/

[77] I. Cabani, G. Toulminet, and A. Bensrhair, “Color-based detection of vehicle lights,” in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 278–283.

[78] M. P. Philipsen, M. B. Jensen, R. K. Satzoda, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, “Night-time drive analysis using stereo-vision for data reduction in naturalistic driving studies,” in *IEEE Intelligent Vehicle Symposium*, 2015.

[79] S. Sivaraman and M. M. Trivedi, “Integrated lane and vehicle detection, localization, and tracking: A synergistic approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 906–917, 2013.

[80] F. Kimura, Y. Mekada, T. Takahashi, I. Ide, H. Murase, and Y. Tamatsu, "Method to quantify the visibility of traffic signals for driver assistance," *IEEJ Transactions on Electronics, Information and Systems*, vol. 130, pp. 1034–1041, 2010.

[81] Y. Jie, C. Xiaomin, G. Pengfei, and X. Zhonglong, "A new traffic light detection and recognition algorithm for electronic travel aid," in *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, 2013, pp. 644–648.

[82] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 127–140.

[83] R. Satzoda and M. Trivedi, "Drive analysis using vehicle dynamics and vision-based lane semantics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 9–18, 2015.

Paper E

Comprehensive Parameter Sweep for Learning-based Detector on Traffic Lights

Morten B. Jensen, Mark P. Philipsen, Thomas B. Moeslund, and Mohan Trivedi

The paper has been published in the
Advances in Visual Computing, 12th International Symposium on Visual Computing, pp. 92–100, 2016.

The layout has been revised.

ABSTRACT

Determining the optimal parameters for a given detection algorithm is not straightforward and what ends up as the final values is mostly based on experience and heuristics. In this paper we investigate the influence of three basic parameters in the widely used Aggregate Channel Features (ACF) object detector applied for traffic light detection. Additionally, we perform an exhaustive search for the optimal parameters for the night time data from the LISA Traffic Light Dataset. The optimized detector reaches an Area-Under-Curve of 66.63 % on calculated precision-recall curve.

1 INTRODUCTION

The Aggregate Channel Features (ACF) object detector [1], from Piotr's Computer Vision Matlab Toolbox (PMT) [2], has been used for detecting a wide range of objects. Originally it was introduced as as a detector for pedestrians in [1], but have since been applied in several other areas related to driver assistant systems (DAS). The applied areas are not only limited to looking-out of the vehicle [3], where other vehicles [4], signs [5], and traffic lights (TLs) [6] have been popular, but also looking-in areas, such as hands detection [7] has seen use of the ACF detector. General for all areas is that the ACF object detector has been adjusted heuristically in a practical manner. Fine-tuning towards the optimal parameters are a common problem amongst researchers as it can be difficult without any prior experience of applying the given detector or without any prior knowledge of the test data. All of the above DAS areas where ACF has been applied are great challenges and remains important cases as people unfortunately keeps getting injured in the traffic. In 2012, 683 people died and 133,000 people were injured in crashes related to red light running in the USA [8]. Traffic light detection is thus an obvious part of DAS system in the transition towards fully autonomous cars.

A large issue in research is that evaluations are done on small and private datasets that are captured by the authors themselves. For better and easier comparison in DAS related areas, benchmarks such as the *VIVA-challenge* [9] and *KITTI Vision Benchmark Suite* [10] can highly beneficial for determine the prone future research directions.

In this paper, we will do a comprehensive analysis of three central parameters for the ACF object detector, applied on the night data from freely available LISA Traffic Light Dataset used in the VIVA-challenge [11]. The contributions of this paper are thus threefold:

1. Exhaustive parameter sweep of ACF.

2. Analysis of correlations between detector parameters.

3. Optimized TL detection results on the night data from the LISA Traffic Light Dataset.

The paper is organized as follows: Relevant previous work is summarized in section 2. In section 3 we present the detector and the three parameters that are investigated. The extensive evaluation of the parameter sweep is presented in 4. Finally, in section 5 we give our concluding remarks.

2 RELATED WORK

The related work can be split into two parts: model-based and learning-based approaches. For a more comprehensive overview of the related work, we refer to [11].

2.1 Model-based

Model-based object detection is a very popular approach for detecting TLs. Most model-based detectors are defined by some heuristic parameters, in most cases relying on color or shape information for detecting TL candidates. The color information is used by heuristically defining thresholds for the color of interest in a given color space [12, 13]. The shape information is usually found by applying circular Hough transform on an edge map [14], or finding circles by applying radial symmetry [15, 16]. In [17, 18] shape information is fused with structural information and additionally color information in [19, 20]. The output of using above approaches are usually a binary image with TL candidates. BLOB analysis is introduced to reduce the number of TL candidates by doing connected component analysis and examining each BLOB by it's size, ratio, circular shape, and so on [21].

2.2 Learning-based

One of the first learning-based detectors is introduced in [22, 23] where a cascading classifier is tested using Haar-like features, but was unable to perform better than their Gaussian color classifier. The popular combination of Histogram of Oriented Gradients (HoG) features and SVM classifier were introduced in [24], but additionally also relying on prior maps with very precise knowledge of the TL locations. The learning-based ACF detector has previously been used for TLs, where features are extracted as summed blocks of pixels in 10 different channels created from the original input RGB frame. In [25] and [6] the extracted features are classified using depth-2 and depth-4 decision trees, respectively. In [6] the octave parameter, which define the number of octaves to compute above the original scale, is changed from 0 to 1.

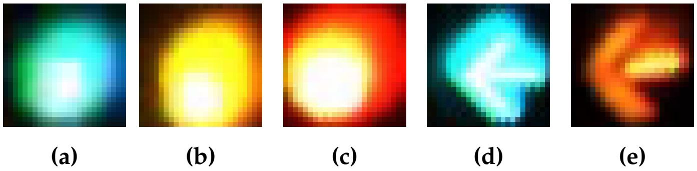


Fig. E.1: Positives samples cropped from training data.

3 METHOD

The method section is two-fold, firstly the learning-based ACF detector is presented. Secondly, the method for conducting the comprehensive parameters optimization for the TL detector is presented.

3.1 Learning-based detector

The features for the ACF object detector are extracted from 10 feature channels: 1 normalized gradient magnitude channel, 6 histogram of oriented gradients channels, and 3 channels constituting the LUV color channels. The features are hence created by single pixel lookups in the feature maps. The channels sub-sampled corresponds to a halving of the dimensions. [4]

The training is done using 3,728 positives TL samples with a resized resolution of 25x25, and 5,772 frames without any TLs and hard negatives generated from 1 execution of bootstrapping on the 5 night training clips from the LISA TL dataset [11]. Examples of these hard negatives are seen in Figure E.2. The number of extracted negative samples varies depending on the configuration, but is limited to maximum of 175,000 samples.

AdaBoost is used to train 3 stages of soft cascades, the three stages consists of 10, 100, and 4000 weakleaners. However, the comprehensive parameters optimization showed that it often converges earlier. The generated AdaBoost classifier is using decision trees as weak learners.

For detecting TLs at greater distances, the intervals of scales can be adjusted by the *octave up* parameter, e.g. changing it from 0 to 1 will define the number of octaves to compute above the original scale. The number of extracted samples from the training will highly depend on the model size, tree-depth, and octave up parameters.

Finally the detection is done by using a sliding window which is moved across each of the 10 aggregated feature channels created from the test frame.

3.2 Parameter optimization

In this paper, a comprehensive parameter optimization is made by adjusting the dimensions of the sliding window, hereafter defined as *mDs*, the decision

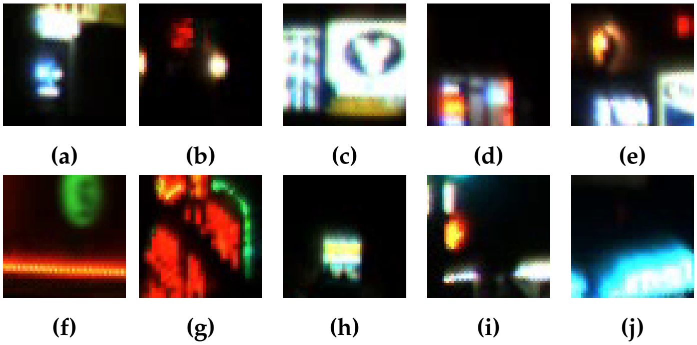


Fig. E.2: Hard negatives generated from bootstrapping.

tree's depth, hereafter defined as *treeDepth*, and the number of octaves to compute above the original scale, hereafter defined as *nOctUp*. To speed up the parameters optimization, a MATLAB script is developed which uses a FTP connection to communicate with a master web host, such n-computers can work on the parameter optimization simultaneously.

The parameter optimization is done by adjusting one parameter at a time, e.g. creating a TL detector with a nOctUp = 0 and treeDepth = 2, and then vary the mDs size from [12,12] to [25,25]. A total of $14^2 = 196$ detectors are made with above nOctUp and treeDepth settings. By adjusting the nOctUp and treeDepth and redoing the sliding window variation, a very comprehensive overview of what the optimal mDs size is, and how the performance correlate with the nOctUp and treeDepth.

4 EVALUATION

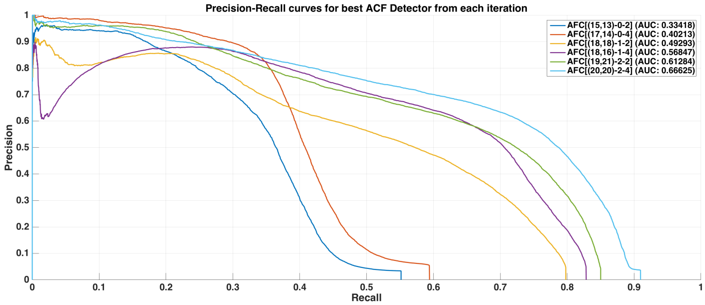


Fig. E.3: PR-curves of best ACF detector from each heatmap.

In this paper the parameters optimization will be done according to the parameter variations seen in Table E.1. The parameters optimization will be performed on nighttime sequence 1 from the LISA TL dataset which are collected in an urban environment in San Diego, USA and contain 4,993 frames and 18,984 annotations. The data is generated from a 5min and 12s long video sequence containing 25 physical TLs split between 5 different types: go, go left, warning, stop, and stop left. [11]

The mDs are decreased in the last two iteration in Table E.1 as the training time increases significantly when the nOctUp and treeDepth are increased. As the training have been done on multiple different computers, the average training time, defined in Table E.1, is calculated from calculated the average training time from the computer being involved in all 6 iterations for the most comparable results. The most involved computer is a Lenovo Thinkpad T550 with an Intel i7-5600U CPU @ 2.6 GHz, 8GB of memory, and a SSD page file. The parameter sweep was done using MATLAB R2015b on Windows 7 Enterprise, both 64-bit.

Table E.1: ACF detector parameter variation

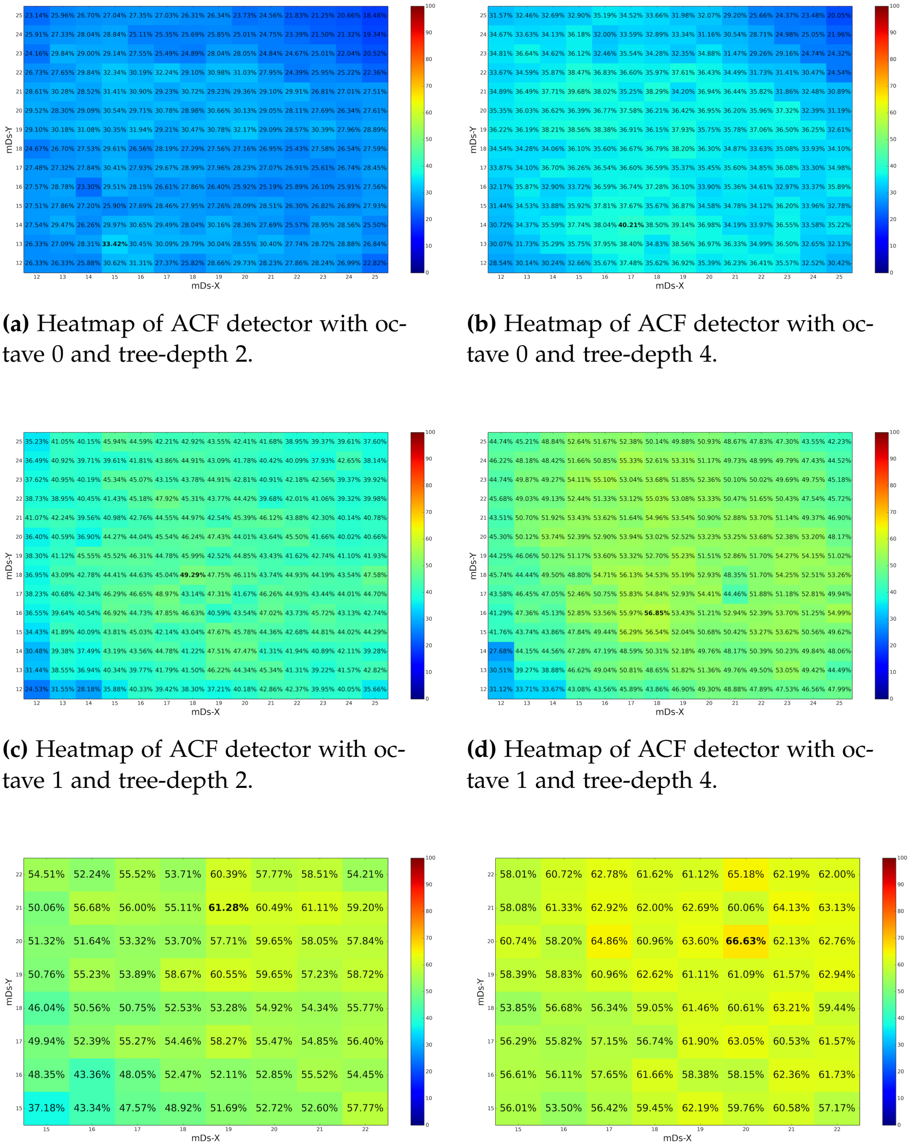
mDs Start	mDs End	nOctUp	treeDepth	# Detectors	Avg Time [DD:HH:MM]
[12, 12]	[25, 25]	0	2	196	00:02:59
[12, 12]	[25, 25]	0	4	196	00:04:40
[12, 12]	[25, 25]	1	2	196	01:02:43
[12, 12]	[25, 25]	1	4	196	01:06:34
[15, 15]	[22, 22]	2	2	64	02:19:28
[15, 15]	[22, 22]	2	4	64	02:21:19
				912	

Each detections will be quantified in accordance to the VIVA-challenge [9], where the Area-Under-Curve(AUC) of a Precision-Recall curve(PR-curve) generated from the ACF results is used as the final evaluation metric [11]. Furthermore, the true positive criteria in the VIVA-challenge defines a detection as one that is overlapping with an annotation with more than 50 %, as defined in Equation (E.1).

$$a_0 = \frac{\text{area}(B_d \cap B_{gt})}{\text{area}(B_d \cup B_{gt})} \tag{E.1}$$

Where a_0 denotes the *overlap ratio* between the detected bounding box B_d and the ground truth bounding box B_{gt}. a_0 must be equal or greater that 0.5 to meet true positive criteria. [26]

In Figure E.4, the 6 different parameter variation sweeps, defined in Table E.1, are seen. All of the heatmaps are plotted with the same color range, spanning from dark blue to dark red indicating a detection rate of 0 % and



(a) Heatmap of ACF detector with octave 0 and tree-depth 2.

(b) Heatmap of ACF detector with octave 0 and tree-depth 4.

(c) Heatmap of ACF detector with octave 1 and tree-depth 2.

(d) Heatmap of ACF detector with octave 1 and tree-depth 4.

(e) Heatmap of ACF detector with octave 2 and tree-depth 2.

(f) Heatmap of ACF detector with octave 2 and tree-depth 4.

Fig. E.4: Heatmaps of ACF Detector with varying octaves and tree-depths.

100 %, respectively. For each heatmap plot in Figure E.4, the model dimension with the highest detection rate is marked with bold. By examining the figures in pairs, e.g. E.4a+E.4b and E.4a+E.4c, one can determine the effect of changing tree-depth or octave, respectively. Increasing only the octave from 0 to 1 increases the best performance from 33.42 % to 49.29 %. Furthermore, the average AUC of the entire heatmap is also increased significantly as a result of the octave increment, which is best illustrated by the increase of more bright green areas in Figure E.4c compared to Figure E.4a. Increasing the tree-depth from 2 to 4 increases the best performing mDs with 6.79 %, and the overall average AUC is also increased by comparing the color schemes of Figure E.4a and E.4b. In Figure E.4d both the octave and tree depth is increased to respectively 1 and 4, resulting in an AUC of 56.85 % with a mDs of [18,16]. There are no clear tendency of a groupings of mDs where the detection rate is good in Figure E.4a. In Figure E.4a, E.4b E.4c, and E.4d, a grouping with a lower detection rate is present in the upper right corner and the lower left corner, which suggests that the optimal mDs is found between a size of 15 and 22. Finally, the octave increased in Figure E.4e and E.4f, where only the detection with mDs from 15 to 22 have been executed due to time restrictions and the previously mentioned low detection rate grouping analysis. Increasing the octave to 2 increases the AUC to 61.28 with a tree-depth 2, and finally 66.63 %, which is the highest achieved AUC in this parameter sweep.

In Figure E.3, the Precision-Recall curves of the best performing mDs from each heatmap are seen. The precision is decent when the recall is under 0.35 for all of the detections, meaning that we have high confidence in our detections until this point. The detections with octave 0 detects less than 60 % of the true positives, by increasing the octaves the recall, and number of true positives detections, are greatly improved reaching over 90 % with octave 2 and tree-depth 4. By increasing the octave all detections reaches a recall above 79 % resulting in a higher AUC.

5 CONCLUSION

Increasing only the octave provides us with better capabilities of detect a larger size range of TLs, resulting in the most significant AUC increments. The increments of the tree-depth improves the results when keeping the octave unchanged, however, the AUC increase is not as high as increasing the octave while keeping the tree-depth the same. The AUC is nearly doubled by increasing both of tree-depth and octave in Figure E.4a and E.4d, leading to conclusion that these parameters are correlated, as the color scheme strongly show the overall AUC increase. Finally, the AUC is improved by increasing octave and tree-depth additionally, as seen in Figure E.4e and E.4f, respectively. As in the first 4 iteration heatmaps, the best performing AUC is increased when increasing both octave and tree-depth simultaneously, which supports the conclusion that the parameters are highly correlated. By examining Figure E.4f it is clear that the best performing AUC is increased additionally and found at a mDs of [20,20] with 2 octaves and a tree-depth of 4.

Further experiments includes finding the convergence points by keep increasing the parameters. Additionally, a similar parameter sweep on the daytime data from the LISA TL dataset would be interesting.

REFERENCES

[1] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *BMVC*, 2010.

[2] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html, 2016.

[3] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, pp. 108–120, 2007.

[4] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[5] A. Mogelmose, D. Liu, and M. M. Trivedi, "Traffic sign detection for us roads: Remaining challenges and a case for tracking," in *IEEE Transactions on Intelligent Transportation Systems*, 2014, pp. 1394–1399.

[6] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection at night: Comparison of a learning-based detector and three model-based detectors," *11th Symposium on Visual Computing*, 2015.

[7] N. Das, E. Ohn-Bar, and M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Sept 2015, pp. 2953–2958.

[8] The Insurance Institute for Highway Safety (IIHS). (2015) Red light running. [Online]. Available: http://www.iihs.org/iihs/topics/t/red-light-running/topicoverview

[9] Laboratory for Intelligent and Safe Automobiles, UC San Diego. (2015) Vision for Intelligent Vehicles and Applications (VIVA) Challenge. http://cvrr.ucsd.edu/vivachallenge/. [Online]. Available: http://cvrr.ucsd.edu/vivachallenge/

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, 2015.

[12] M. Diaz-Cabrera, P. Cerri, and P. Medici, "Robust real-time traffic light detection and distance estimation using a single camera," *Expert Systems with Applications*, pp. 3911–3923, 2014.

[13] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, "Night-time traffic light detection based on svm with geometric moment features," *World Academy of Science, Engineering and Technology 76th*, pp. 571–574, 2013.

[14] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 809–812.

[15] G. Siogkas, E. Skodras, and E. Dermatas, "Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information." in *VISAPP (1)*, 2012, pp. 620–627.

[16] S. Sooksatra and T. Kondo, "Red traffic light detection using fast radial symmetry transform," in *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2014, pp. 1–6.

[17] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "Tracking both pose and status of a traffic light via an interacting multiple model filter," in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[18] R. Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 333–338.

[19] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, "A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles," in *33rd Chinese Control Conference (CCC)*, 2014, pp. 4924–4929.

[20] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 707–723, 2012.

[21] D. Nienhuser, M. Drescher, and J. Zollner, "Visual state estimation of traffic lights using hidden markov models," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1705–1710.

[22] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich, "Making bertha see," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 214–221.

[23] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 49–53.

[24] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[25] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," *18th IEEE Intelligent Transportation Systems Conference*, 2015.

[26] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

Paper F

Evaluating State-of-the-art Object Detector on Challenging Traffic Light Data

Morten B. Jensen, Kamal Nasrollahi, and Thomas B. Moeslund

The paper has been published in the
Computer Vision and Pattern Recognition Workshops: Traffic Surveillance Workshop and Challenge

ABSTRACT

Traffic light detection (TLD) is a vital part of both intelligent vehicles and driving assistance systems (DAS). General for most TLDs is that they are evaluated on small and private datasets making it hard to determine the exact performance of a given method. In this paper we apply the state-of-the-art, real-time object detection system You Only Look Once, (YOLO) on the public LISA Traffic Light dataset available through the VIVA-challenge, which contain a high number of annotated traffic lights, captured in varying light and weather conditions.

The YOLO object detector achieves an AUC of impressively 90.49 % for daysequence1, which is an improvement of 50.32 % compared to the latest ACF entry in the VIVA-challenge. Using the exact same training configuration as the ACF detector, the YOLO detector reaches an AUC of 58.3 %, which is in an increase of 18.13 %.

1 INTRODUCTION

In recent years the term *big data* and *machine learning* have gained tremendous momentum, especially the use of big data have been a heavily discussed topic. As a result, data is collected in almost every digital action we do, and is collected like never before. In fact, we create 2.5 quintillion bytes (2,500,000,000 gigabytes) of data each day resulting in 90 % of the current available data have been created for the past 2 years [1]. The data are collected from a large variety of locations, spanning from your social media activities and browsing to various sensors collecting climate data or traffic surveillance data. Collecting traffic data both with the purpose of surveillance and especially autonomous vehicles have gained a lot of media attention as a result of major companies spending large amount money on research in this area. However, making a vehicle drive autonomously have a lot of challenges linked to it, which still requires years of research.

Both industry and academic institutions are looking into research and applications that can be relevant and helpfull in the meantime. This can prove beneficial for the ultimate dream of self-driving cars, but also for the popular driving assistance systems (DAS). DAS applications are already widely implemented in newer vehicles, such as emergency breaking, automatic lane changing, keeping the advertised speed limit, and adaptive cruise control. DAS applications can usually be split into looking-in [2], such as hands activity recognition [3] and looking-out applications, such as detection of other vehicles, pedestrians [4], traffic signs [5] or traffic lights [6]. In 2012, 683 people died and 133,000 people were injured in crashes related to red light running in the USA [7], making traffic light detection a vital part of both self-driving cars and DAS.

In this paper we apply the state-of-the-art, real-time object detection system *You Only Look Once*, (YOLO) [8], which have proven a good competitor to Fast R-CNNs and SSDs both in terms of detections and speed. In this paper, we will apply YOLO on the daytime data from the freely available LISA Traffic Light Dataset used in the VIVA-challenge [9, 10], which have seen a limited use of deep learning methods. The contributions of this paper is twofold:

- Training and applying the state-of-the-art, real-time object detection system *You Only Look Once*, (YOLO) for traffic light detection.
- Deep learning entry in the public VIVA Traffic Light challenge.

The paper is organized as follows: Relevant research is summarized in section 2. In section 3 we present the method used, followed by evaluation of the TL detector in section 4. Finally, section 5 rounds of with some concluding remarks.

2 RELATED WORK

In this section a brief introduction to the most notable research in relation to TLD is given, for a more comprehensive overview, we refer to the traffic light survey [9]. In [9] TLD is split into two categories: model-based and learning-based.

The model-based methods have been quite dominant and popular in the past decade and are usually created by the use of a heuristically defined model which relies on color and/or shape information. The color information is quite intuitive and a straight-forward approach as traffic lights presents the driver with multiple color cues which corresponds to a driver action e.g. stop or go. The detector is based on a heuristical defined threshold in a selected color space [11, 12]. The color can however vary from scene to scene and thus challenge models relying solely on static thresholds. So rather than looking at color, one could make use of the distinctive shape of traffic lights by applying circular Hough transform on an edge map [13] or by using radial symmetry [14]. Both approaches are challenged in different scenarios, but not entirely the same scenarios, thus shape information is fused with structural information [15, 16], and additionally color information in [17, 18]. Rather than defining static set of rules, [19] propose a Bayesian inference framework relying on color, shape and height to detect traffic lights.

Cascading classifier based on Haar-like features was one of the first learning-based detectors to be introduced in [20, 21], but did however not outperform their Gaussian color classifier. As for most other computer vision research areas, the popular combination of using Histogram of Oriented Gradients features together with a SVM classifier was introduced in [22]. The learning-based Aggregated Channel Features (ACF) detector have seen a large use in TLD, and have shown superior performance over the heuristic models both during day and night time [6, 23]. TLD using Convolutional Neural Network (CNN) is introduced in [24, 25], where a CNN is used detects and recognize the traffic lights using region-of-interest information provided by an onboard GPS sensor.

3 METHOD

In this section the method used in this paper will be briefly introduced.The method section is split into two sections: firstly the YOLO object detector is introduced. Secondly, training parameters and data specifications used in the evaluation are introduced.

3.1 YOLO

YOLO have been introduced in two versions [8, 26], where the latest version is the one used in this paper which include new features as well as modifications to the existing network. YOLO is an end-to-end single convolutional neural network that detects objects based on bounding boxes prediction and class probabilities. The network divides the input image into a SxS grid, if the center of an object is located within this grid, it is this specific grid's task to detect the object. Each grid predicts bounding boxes and a corresponding confidence, where the confidence is an indicator of how confident the model is that a box contains an object as well as how accurate the box is. The confidence is therefore calculated using the intersection over union (IOU), where a perfect match between a predicted box and a ground truth will provide a confidence of 1, and oppositely if a predicted box is not present in the grid, hence no ground truth overlapping, the confidence will be 0. Finally, the grid cell also predicts the probability of an object belonging to a class.

Unlike many sliding window methods, such as the ACF detector, YOLO examines the entire image during training helping it to learn contextual information about a given class and its surroundings. The original YOLOv2 classification model, called Darknet-19, has 19 convolutional layers and 5 maxpooling layers, and have some resembles to well-known VGG-16 network. It is however a lot less complex as the VGG-16 requires 30.69 billion floating point operations to process a single 224x224 pixel frame, whereas the Darknet-19 only needs 5.58 billion operations whilst improving the top-5 accuracy on ImageNet with 1.2 % compared to VGG-19's 90 %. An additional training where the size is increased from 224 to 448, improves the top-5 accuracy to 93.3 % at the compromise of processing the images 4.24 times slower.

This 448x448 model constitutes the Darknet19 448x448 model which have been used as pre-weights for training in this paper.

For using the model for detection, the network is modified by removing the last convolutional network and instead adding three 3x3 convolutional layers with 1024 filters, which is finally followed by a 1x1 convolutional layer with the number of outputs needed for the specific detection. For enabling fine grain features, a passthrough layer is inserted second to the last convolutional layer.

3.2 Training parameters

The *random* parameter enables multi-scale training, resulting in a robustness for detecting objects in different image resolutions. The input size is per default set to a resolution of (416x416), but by enabling the random parameter the network will randomly change the input image size every 10 batch. The YOLOv2 network downsamples by a factor of 32, resulting in a downsampling range between {320, 352, ..., 608}. The smallest input size is thus (320x320), and the largest input size is (608x608). The random parameter is per default enabled in YOLOv2, in this paper we will try to identify the effect. Furthermore, we will investigate varying the input size whilst doing detection.

4 EVALUATION

Several models have been trained using different training data and modified in accordance to the parameters described in section 3.2.

The data configuration for each model can be seen in Table F.1. The training data used for all the models are from the LISA Traffic Light Dataset [9] and the LARA Traffic Light Dataset [27].

The LISA Traffic Light Dataset consists of 13 day training clips, hereafter referred to as LISA-dayTrain, as well as 2 longer test sequences, hereafter referred as LISA-daySeq1 or 2. For evaluating, the LISA-daySeq1 has been used, as it was the main evaluation sequence in the VIVA-challenge. The LARA Traffic Light Dataset is also included to create some more variance as it is captured in Paris, France, whereas the LISA TL dataset is captured in San Diego, USA. Furthermore the LARA Traffic Light Dataset is introduced to see how it impacts the model when testing it on a test sequence that is captured in same environment as a large part of the training data. In Table F.2, an overview of the used training and test data is seen. In Figure F.1 some samples from the data are seen.

A total of 6 YOLO TLD models are trained and applied on the LISA-daySeq1. In order to make the results of above models comparable with pre-

Table F.1: Overview of the trained YOLOv2 Traffic Light Detectors. All models have been trained with an input image size of (416,416), with half the models enabled the random parameter varying the input image size between {320, 352, ..., 608}.

		Training Data		
Model name	*Random*	*LISA-dayTrain*	*LARA [27]*	*LISA-daySeq2*
YOLO_V1_0		✓		
YOLO_V1_1	✓	✓		
YOLO_V2_0		✓	✓	
YOLO_V2_1	✓	✓	✓	
YOLO_V3_0		✓		✓
YOLO_V3_1	✓	✓		✓

Table F.2: Overview of the evaluation data.

Dataset	Frames	True positives	Resolution	Classes
LARA	11,179	9,168	640 x 480	4 (green, orange, red, & ambiguous)
LISA-dayTrain	14,025	40,764	1280 x 960	6 (Go, go left, warning, warning left, stop, stop left)
LISA-daySeq2	6,894	11,144	1280 x 960	6 (Go, go forward, go left, warning, stop, stop left)
LISA-daySeq1	4,060	10,308	1280 x 960	5 (Go, warning, warning left, stop, stop left)

vious publications, the results must be evaluated in accordance to the VIVA-challenge [10], where the Area-Under-Curve(AUC) of a Precision-Recall curve(PR-curve) is the final evaluation metric [9]. Furthermore, the true positive criteria in the VIVA-challenge defines a detection as one that is overlapping with an annotation with more than 50 %, as defined in Equation (F.1).

$$a_0 = \frac{\text{area}(B_d \cap B_{gt})}{\text{area}(B_d \cup B_{gt})} \tag{F.1}$$

Where a_0 denotes the *overlap ratio* between the detected bounding box B_d and the ground truth bounding box B_{gt}. a_0 must be equal or greater than 0.5 to meet true positive criteria. [28]

Prior to calculating the AUC of the model, we examine the recall of each of the trained models. Models are trained for 80,000 iterations and for every 1000th iteration during training, weights are saved for backup purposes. These weights are used to determine how the performance relates to the number of iterations. This relation is seen Figure F.2 and in F.3 where the

(a) (b) (c)

(d) (e) (f)

Fig. F.1: Training samples from the (a-d) LISA and (e-f) LARA Traffic Light database.

detectors' image size have been changed from (416,416) to (672,672).

In Figure F.2 the detectors with an input image of (416,416) are shown. To determine the impact of the random parameter, we compare the versions of the YOLO TL detectors. By enabling the random parameter with only the LISA-dayTrain as training data, the recall performance decrease by 17.32 %. By examining the figure, it is clear that YOLO_V1_1 is struggling to reach a stable recall compared to the other 5 models, which suggests that we do not use enough and sufficient varied training data for the varying input image size to make any impact. In YOLO_V2_0 we add the LARA dataset to the training which nearly reaches the same recall as YOLO_V1_0. YOLO_V2_1, with the random parameter enabled, increases the recall with 3.85 % compared to YOLO_V2_0 but is still 2.47 % worse than YOLO_V1_0. Finally, by swapping the LARA dataset with LISA-daySeq2, we reach a recall of 87.38 % and 88.91 % for YOLO_V3_0 and YOLO_V3_1, respectively.

As the detectors only use convolutional and pooling layers we can resize the input image size without retraining. In Figure F.5 the detectors with input image of (672,672) are shown. The result of increasing the input image size to (672,672) provides a very similar picture of the detectors as for the (416,416). However, 5 out of 6 models reaches a higher maximum recall after increasing the image input size to (672,672), the exception being YOLO_V1_1 which also struggled in Figure F.2. By examining and comparing Figure F.2 and F.3, it is clear from a visual analysis, that the (416,416) looks more smooth compared to (672,672). This is due to the larger difference in the recall results between the iterations, suggesting that the input image size of (672,672) might not be

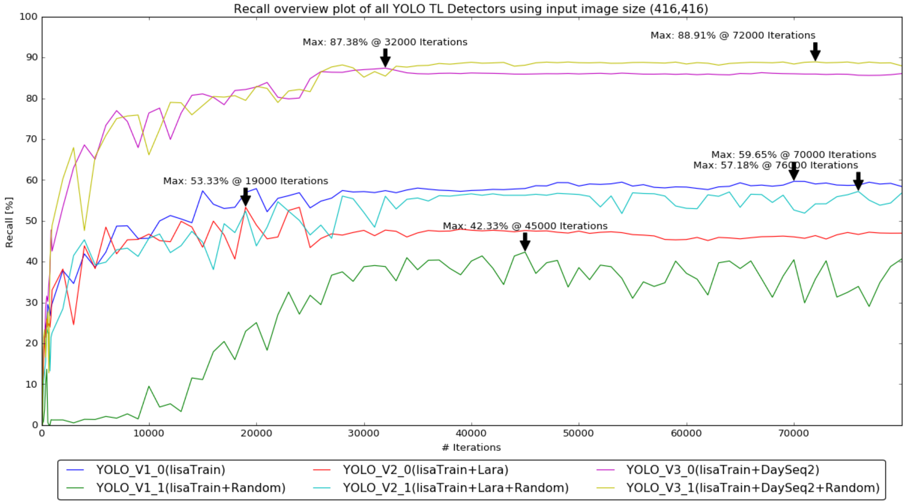


Fig. F.2: Recall plot for iterations made during training of the models with input image size (416,416).

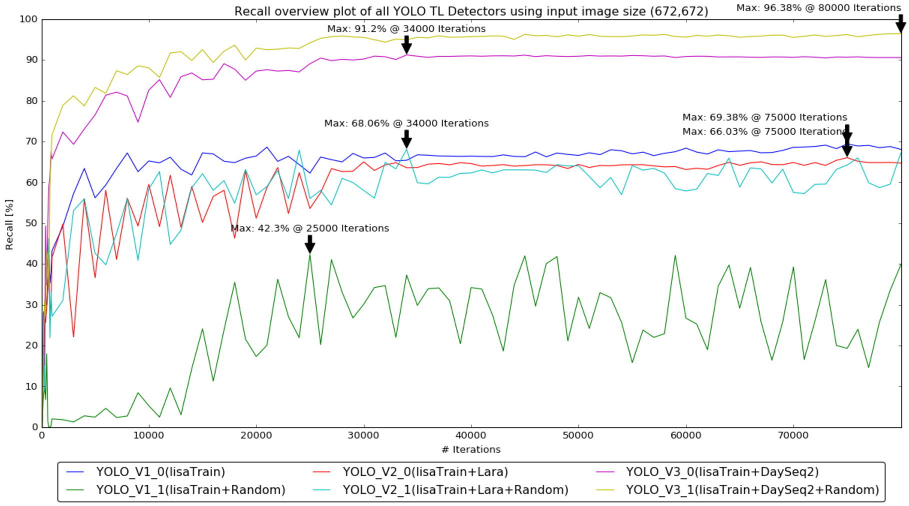


Fig. F.3: Recall plot for iterations made during training of the models with input image size (672,672).

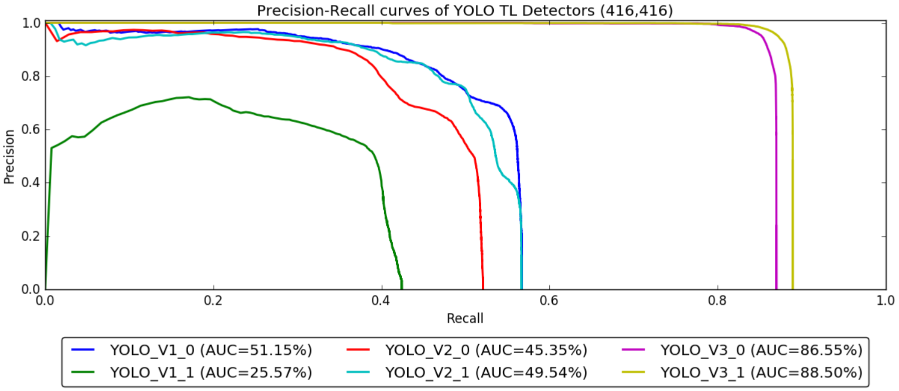


Fig. F.4: Precision-recall curves of the best recall iterations from (416,416) detectors in Figure F.2.

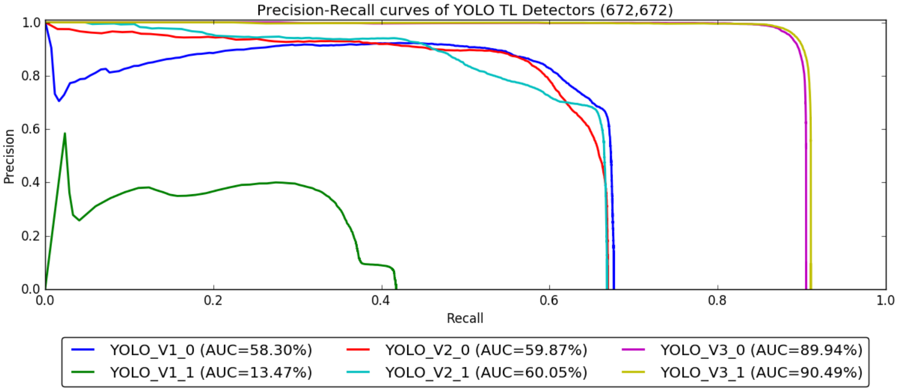
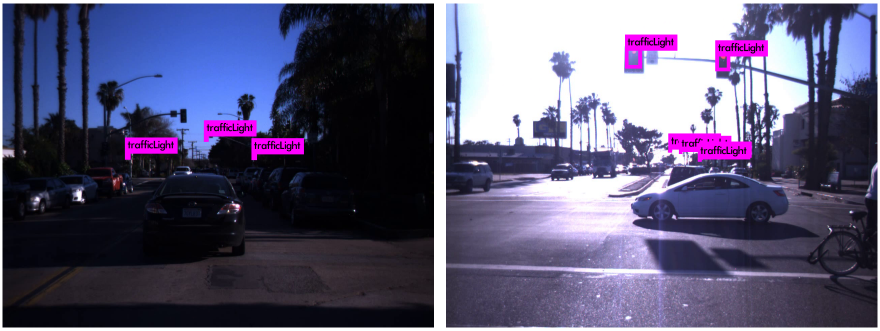


Fig. F.5: Precision-recall curves of the best recall iterations from (672,672) detectors in Figure F.3.



(a) **(b)**

Fig. F.6: Results from YOLO V3 1 applied on LISA-daySeq1.

completely ideal, at least not for the data configuration of YOLO_V1 and V2. Finally, the best performing model is YOLO_V3_1, which was expected as it is the one with the most training data from the LISA TL dataset, thus looking most identical with LISA-daySeq1.

For each of the detectors seen in Figure F.2 and F.3, the iteration with the highest recall is used for precision-recall curves and calculating the corresponding AUC. The lowest AUC is in both figures the YOLO_V1_1, which is not surprising as it was also generally performing bad in Figure F.2 and F.3. In Figure F.4, the YOLO_V1_0 is reaching an AUC of 51.51 % and is the best performing of the one not including LISA-daySeq2 in the training data. In Figure F.5, YOLO_V1_0 is still performing good, but both YOLO_V2_0 and YOLO_V2_1 surpass it after the image input size is increased. Generally increasing the input image size provided an average AUC increase of 4.29 %, and if we exclude the YOLO_V1_1 we each an average AUC increase of 7.51 %. The average AUC increase caused by enabling the random parameter for YOLO_V2 and YOLO_V3 is 1.72 %, which could indicate that adjusting the input image size provide a larger impact.

The 2 detectors based on both LISA-dayTrain and LISA-daySeq2, YOLO_V3_0 and YOLO_V3_1, reaches the by far highest AUC with both image input sizes. The highest overall AUC is 90.49 % by YOLO_V3_1. In [29], the highest AUC for daySeq1 is 40.17 %, which means that the YOLO_V3_1 have significantly improved the entry on the LISA Traffic Light dataset with impressively 50.32 %. This result do however not form basis for a fair comparison between YOLO and the ACF detector used in [29] as the ACF detector have purely been trained on the lisaTrain data. So to compare the performance of the two methods given the same data, we must compare it to YOLO_V1_0 which reaches an AUC of 58.3 % with an image input size of (672,672), resulting in an AUC increase of 18.13 %.

In Figure F.6, detection results from the YOLO_V3_1 detector are shown. Compared to previous work from the ACF detector used in [29], the YOLO_V3_1 handles the varying lighting conditions well as seen from F.6a. Generally, the models with the multi-scale training parameter *random* enabled are not surprisingly also able to detect the TLs at a much longer distance, which is illustrated in F.6b.

5 CONCLUSION

We have taken one of the state-of-the-art object detectors and applied in on a challenging traffic light dataset with different model and data configurations. The highest overall AUC on daySequence1 from the LISA Traffic Light dataset is 90.49 % and is unsurprisingly based on all the training data and daySequence2 from the same dataset. This improves the entry from [29] on the LISA Traffic Light dataset with impressively 50.32 %. However, if we use the exact same training data as used with the ACF detector in [29], we reach an AUC of 58.3 %, which is an AUC improvement of 18.13 %. The random parameter that enables multi-scale training did in most cases improve the AUC slightly, whereas increasing the input image size of the detector turned out to have a larger impact than the random parameter.

Further experiments includes using SSD for traffic light detection, creating an ensemble with R-FCN, and do similar evaluation on the nighttime data from the LISA Traffic Light dataset.

REFERENCES

[1] R. H. Bajaj and P. Ramteke, "Big data–the new era of data," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 1875–1885, 2014.

[2] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, pp. 108–120, 2007.

[3] E. Ohn-Bar and M. Trivedi, "In-vehicle hand activity recognition using integration of regions," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1034–1039.

[4] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[5] A. Mogelmose, D. Liu, and M. M. Trivedi, "Traffic sign detection for us roads: Remaining challenges and a case for tracking," in *IEEE Transactions on Intelligent Transportation Systems*, 2014, pp. 1394–1399.

[6] M. B. Jensen, M. P. Philipsen, T. B. Moeslund, and M. Trivedi, *Comprehensive Parameter Sweep for Learning-Based Detector on Traffic Lights*. Cham: Springer International Publishing, 2016, pp. 92–100. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-50832-0_10

[7] The Insurance Institute for Highway Safety (IIHS). (2015) Red light running. [Online]. Available: http://www.iihs.org/iihs/topics/t/red-light-running/topicoverview

[8] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[9] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, July 2016.

[10] Laboratory for Intelligent and Safe Automobiles, UC San Diego. (2015) Vision for Intelligent Vehicles and Applications (VIVA) Challenge. [Online]. Available: http://cvrr.ucsd.edu/vivachallenge/

[11] M. Diaz-Cabrera, P. Cerri, and P. Medici, "Robust real-time traffic light detection and distance estimation using a single camera," *Expert Systems with Applications*, pp. 3911–3923, 2014.

[12] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, "Night-time traffic light detection based on svm with geometric moment features," *World Academy of Science, Engineering and Technology 76th*, pp. 571–574, 2013.

[13] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 809–812.

[14] S. Sooksatra and T. Kondo, "Red traffic light detection using fast radial symmetry transform," in *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2014, pp. 1–6.

[15] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "Tracking both pose and status of a traffic light via an interacting multiple model filter," in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[16] R. Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 333–338.

[17] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, "A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles," in *33rd Chinese Control Conference*, 2014, pp. 4924–4929.

[18] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 707–723, 2012.

[19] S. Hosseinyalamdary and A. Yilmaz, "A bayesian approach to traffic light detection and mapping," *{ISPRS} Journal of Photogrammetry and Remote Sensing*, vol. 125, pp. 184 – 192, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092427161730028X

[20] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. Herrtwich, "Making bertha see," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 214–221.

[21] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 49–53.

[22] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[23] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection at night: Comparison of a learning-based detector and three model-based detectors," *11th Symposium on Visual Computing*, 2015.

[24] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in *IEEE 17th International Conference on Intelligent Transportation Systems*, 2014, pp. 2286–2291.

[25] V. John, K. Yoneda, Z. Liu, and S. Mita, "Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching," *IEEE Transactions on Computational Imaging*, vol. 1, no. 3, pp. 159–173, Sept 2015.

[26] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[27] Robotics Centre of Mines ParisTech. (2015) Traffic lights recognition (tlr) public benchmarks. [Online]. Available: http://www.lara.prd.fr/benchmarks/trafficlightsrecognition

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[29] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," *18th IEEE Intelligent Transportation Systems Conference*, 2015.

Paper G

Traffic Light Detection at Night: Comparison of a Learning-based Detector and three Model-based Detectors

Morten B. Jensen, Mark P. Philipsen, Chris Bahnsen, Andreas Møgelmose, Thomas B. Moeslund, and Mohan M. Trivedi

The paper has been published in the
Advances in Visual Computing, 11th International Symposium on Visual
Computing, pp. 774–783, 2015.

The layout has been revised.

ABSTRACT

Traffic light recognition (TLR) is an integral part of any intelligent vehicle, it must function both at day and at night. However, the majority of TLR research is focused on day-time scenarios. In this paper we will focus on detection of traffic lights at night and evaluate the performance of three detectors based on heuristic models and one learning-based detector. Evaluation is done on night-time data from the public LISA Traffic Light Dataset. The learning-based detector outperforms the model-based detectors in both precision and recall. The learning-based detector achieves an average AUC of 51.4 % for the two night test sequences. The heuristic model-based detectors achieves AUCs ranging from 13.5 % to 15.0 %.

1 INTRODUCTION

Traffic lights are used to safely regulate the traffic flow in the current infrastructure, they are therefore a vital part of any intelligent vehicle, whether it is fully autonomous or employ Advanced Driver Assistance Systems (ADAS). In either application, TLR must be able to perform during both both day and night. TLR for night-time scenarios is especially important as more than 40 % of accidents at intersections occur during the late-night/early-morning hours, in fact a crash is 3 times more probable during the night than during the day [1]. For more introduction to TLR in general we refer to [2] where an overview is given of the current state of TLR. In the same paper, the lack of a large public dataset is addressed with the introduction of the LISA Traffic Light Dataset, which contains challenging conditions and both day- and night-time data.

Before the state of traffic lights (TLs) can be determined they must first be detected. Traffic light detection (TLD) has proven to be very challenging under sub-optimal and changing conditions. The purpose of this paper is therefore to evaluate the night-time TLD performance of three heuristic TL detectors and compare this to a state-of-the-art learning based detector relying on Aggregated Channel Features (ACF). The same learning-based detection framework has previously been applied for day-time TLD in [3]. This makes it possible to compare the detector's performance at night and day. Evaluation is done on night-time sequences from the extensive and difficult LISA Traffic Light Database. The contributions are thus threefold:

1. First successful application of a state-of-the-art learning-based detector for TLD at night.

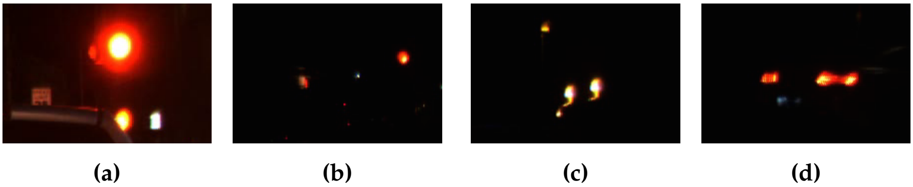
2. Comparison of three model-based TLD approaches and a learning-based detector using ACF.

3. Clarification of the challenges for night-time TLD.

The paper is organized as follows: Challenges specific to night-time TLD are clarified in section 2. Relevant research is summarized in section 3. In section 4 we present the detectors, followed by the evaluation in section 5. Finally, section 6 rounds of with our concluding remarks.

2 TRAFFIC LIGHTS AND THEIR VARIATIONS

In this section we present some challenges particular to night-time TLD.



(a) **(b)** **(c)** **(d)**

Fig. G.1: Challenges samples from the LISA Traffic Light dataset.

1. Lights may seem larger than the actual source [4], see Figure G.1a.
2. Colors saturate to white [4], see Figure G.1a.
3. Lack of legal standards for tail-lights in the USA, tail-lights may therefore resemble TLs [5], see Figure G.1d.
4. TL may be reflected in reflective surfaces, e.g. storefronts, see Figure G.1b.
5. Street lamps and other light sources may look similar to TLs, see Figure G.1c.

Type 1 and 2 can be reduced by increasing the shutter speed at the risk of getting underexposed frames. One solution to this problem is seen in [6], where frames are captured by alternating between slow and fast shutter speed. Generally, it is hard to cope with the remaining issues from a detection point of view. One solution to removing type 3, 4 and 5 false positives could be the introduction of prior maps with information of where TLs are located in relation to the ego-vehicles location, as e.g. seen in [5].

3 RELATED WORK

Most research on TLD and TLR has been focused on day-time, only a handful of publications evaluate their systems on night-time data. One is [4] where a fuzzy clustering approach is used for detection. Gaussian distributions are calculated based on the red, amber, green, and black clusters in a large number of combinations of the RGB and RGB-N image channels. In [7] the work from [4] is expanded, by the introduction of an adaptive shutter and gain system, advanced tracking, distance estimation, and evaluate on a large and varied dataset with both day-time and night-time frames. Because of the differences in light conditions between night and day, they use one fuzzy clustering process for day conditions and another for night conditions. [8] finds TL candidates by applying the color transform proposed in [9]. The color transform determines the dominant color of each pixel based on the RGB values. Dominant color images are only generated for red and green, since no transform is presented for yellow. After thresholding of the dominant color images, BLOBs are filtered based on the width to height ratio and the ratio between the area of the BLOB and the area of the bounding box. The remaining TL candidates are then classified using SVM on a wide range of BLOB features.

When looking at TL detectors which have been applied to day-time data, two recent papers have employed learning-based detectors. [10] is combining occurrence priors from a probabilistic prior map and detection scores based on SVM classification of Histogram of Oriented Gradients (HoG) features to detect TLs. [3] uses the ACF framework provided by [11]. Here features are extracted as summed blocks of pixels in 10 channels created from transformations of the original RGB frames. The extracted features are classified using depth-2 learning trees. Spotlight detection using the white top hat operation on intensity images is seen in [12–14] and [15]. In [16], the V channel from the HSV color space is used with the same effect. A high proportion of publications use simple thresholding of color channels in some form. [6] is a recent example where traffic light candidates are found by setting fixed thresholds for red and green TL lamps in the HSV color space.

For a more extensive overview of the TLR domain, we refer to [2].

4 METHODS

In this section we present the used methods. In the first subsection the learning-based detector is described. The second describes each of the three model-based detectors and how the confidence scores are calculated for the TL candidates found by these model-based detectors.

4.1 Learning-based detection

In this subsection we describe how the successful ACF detector has been applied to the night-time TL detection problem. The learning-based detector is provided as part of the Matlab toolbox from [11]. It is similar to the detectors seen in [17] for traffic signs and [3] for day-time TLs, except for few differences in the configuration and training which are described below.

Features

The learning-based detector is based on features from 10 channels as described in [18]. A channel is a representation of the input image, which is obtained by various transformations. The 10 different channels include 6 gradient histogram channels, 1 for unoriented gradient magnitude, and 3 for each channels in the CIE-LUV color space. In each channel, the sums of small blocks are used as features. These features are evaluated using a modified AdaBoost classifier with depth-4 decision trees as weak learners.

Training

Training is done using 7,456 positive TL samples with a resolution of 25x25 and 163,523 negative samples from 5,772 selected frames without TLs. Figure G.2 shows four examples of the positive samples used for training the detector. Similarly, Figure G.3 shows two examples of frames used for negative samples. Finally, Figure G.4 shows four hard negative samples extracted using false positives from the training dataset.

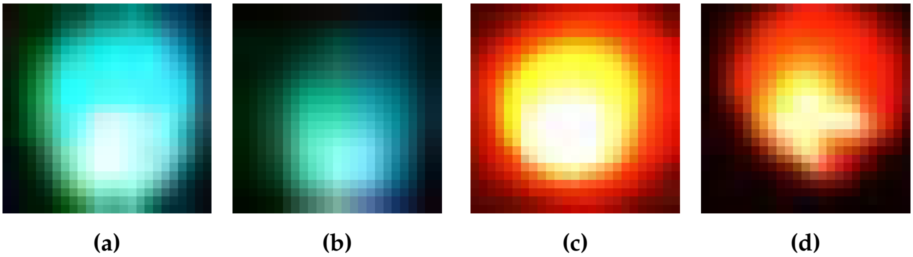
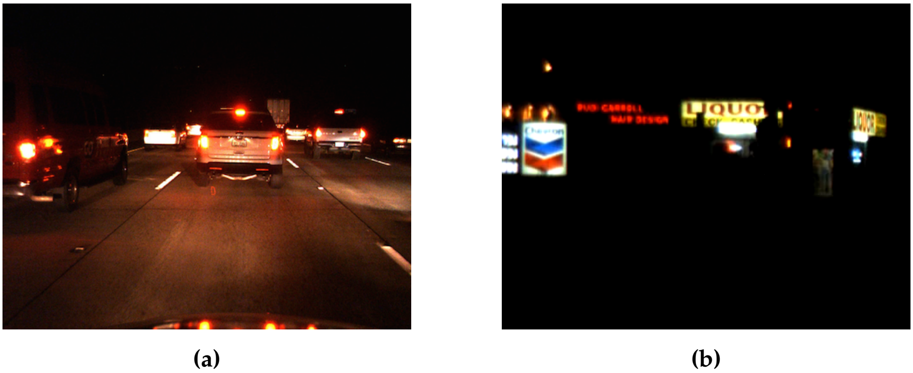


Fig. G.2: Positive samples for training the learning-based detector.

AdaBoost is used to train 3 cascade stages, 1st stage consists of 10 weak learners, 2nd stages of 100, and 3rd stage is set to 4,000 but converges at 480. In order to detect TLs at a greater interval of scales, the octave up parameter is set to 1 instead of the default 0. The octave up parameters defines the number of octaves to compute above the original scale.

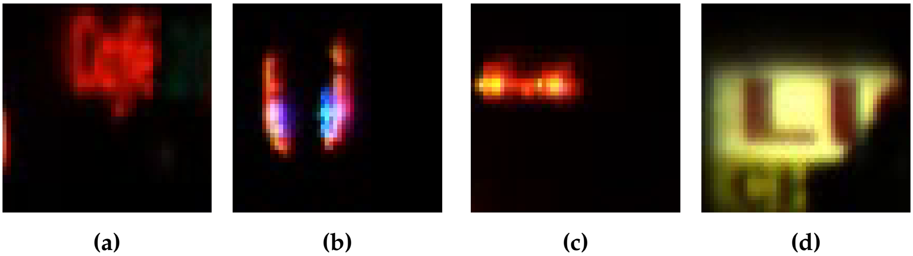
Detection

A 18x18 sliding window is used across each of the 10 aggregated channels in the frames from the test sequences.



(a) (b)

Fig. G.3: Negative samples for training the learning-based detector.



(a) (b) (c) (d)

Fig. G.4: Hard negative samples for training the learning-based detector.

4.2 Heuristic model-based detection

We want to compare the learning-based detector to more conventional detector types which are based on heuristic models. For each of the three model-based detectors, a short description is given along with output showing central parts of the detectors. The sample shown in Figure G.1a is used as input.

Detection by Thresholding

The detector which uses thresholding is mainly based on the work presented in [6]. Thresholds are found for each TL color in the HSV color space by looking at values of individual pixels from TL bulbs sampled from the training clips in the LISA Traffic Light dataset. Figure G.5 (a) shows the input sample and Figure G.5 (b) shows output after thresholding input. Pixels that fall inside the thresholds for one of the three colors are labeled green, yellow or red in Figure G.5. For the input sample only pixels which fell within the yellow and red thresholds were present.

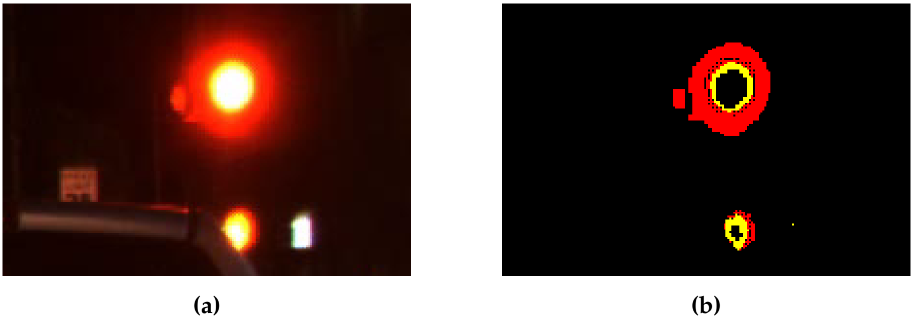


Fig. G.5: Thresholded TL.

Detection by Back Projection

Back projection begins with the generation of color distribution histograms. The histograms are two-dimensional and are created for each of the TL colors using 20 training samples for each of the TL colors, green, yellow, and red. From the training samples the U and V channels of the LUV color space are used. The histograms are normalized and used to generate a back projection which is thresholded to remove low probability pixels from the TL candidate image. The implementation is similar to our previous work in [3]. Figure G.6a shows the back projected TL candidate image. Figure G.6b shows the processed back projected TL candidate image after removal of low probability pixels and some typical morphology operations.

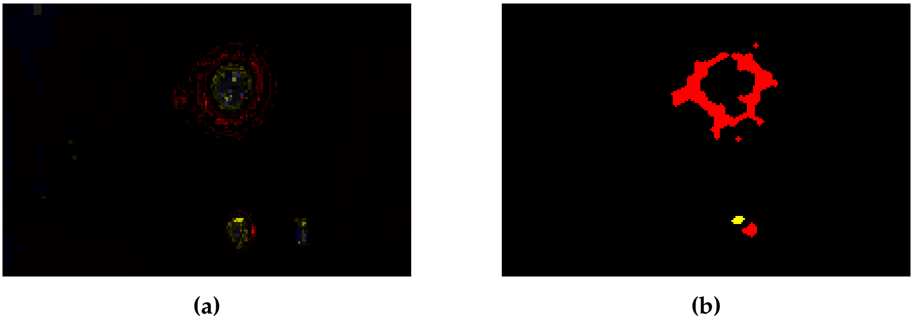
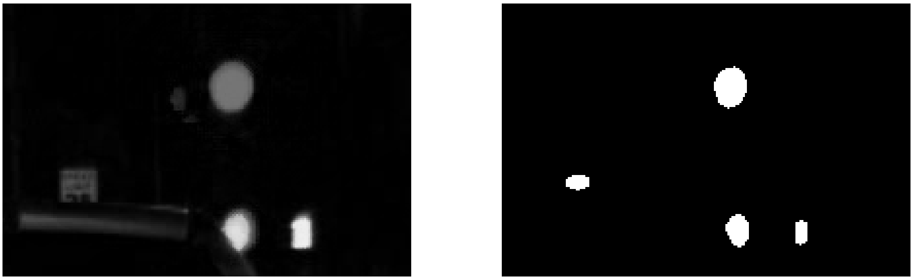


Fig. G.6: Back projected TL.

Detection by Spotlight Detection

Spotlights are found in the intensity channel L from the LUV colorspace using the white top-hat morphology operation. The implementation is similar to our previous work in [3]. This method has been used in a many recent TLR papers [12–16]. Figure G.7a shows the output of the white top-hat operation. Figure G.7b shows the binarized TL candidate image after thresholding and some typical morphology operations.



(a) **(b)**

Fig. G.7: Spotlights found using the white top-hat operation.

Confidence scores for TL candidates

Confidence scores are calculated for all TL candidates found by the three model-based detectors. The TL BLOB characteristics used in this work have seen use in earlier work, such as [9] and [8]. Scores from individual characteristics are generated ranging from [0 - 1], with 1 being the best. These are summed for each TL candidate, resulting in a combined score ranging from [0 - 5].

Bounding box ratio: The bulbs of TLs are circular, therefore under ideal conditions the bounding box will be quadratic. The bounding box ratio is calculated as the ratio between width and height of the bounding box.

Solidity ratio: Since TL bulbs are captured as circular and solid under ideal conditions, a BLOBs solidity is a characteristic feature for a TL. The solidity is calculated as the ratio between the convex area of detected BLOBs and the area of a perfect circle, with a radius approximated from the dimensions of the BLOB.

Mean BLOB intensity: Each of the three detectors produce an intensity channel which can be interpreted as a confidence map of TL pixels. The best example is from detection by back projection, where the result of the back projection is an intensity channel with normalized probabilities of each pixel being a TL pixel. The intensity channel employed from the spotlight detector is less informative, since it describes the strength of the spotlight. From the threshold based detector, we simply use the intensity channel from the LUV color space.

Flood-filled area ratio: The bulbs of TLs are surrounded by darker regions, by applying flood filling from a seed inside the found BLOBs, it can be confirmed that this contrast exists. We use the ratio between the area of the bounding box and the area of the bounding box from the flood filled area as a measure for this.

Color confidence: Using basic heuristics based thresholding we find the most prominent color inside the TL candidates' bounding boxes. The confidence is calculated based on the number of pixels belonging to that color and the total number of pixels within the bounding box. Pixels with very low saturation are not included in the confidence calculation.

5 EVALUATION

Most TL detectors have been evaluated on datasets which are unavailable to the public. This makes it difficult to determine the quality of the published results and compare competing approaches. We strongly advocate that evaluation is done on public datasets such as the LISA Traffic Light Dataset[1].

5.1 LISA Dataset

The four detectors presented in this paper are evaluated on the two night test sequences from the LISA Traffic Light Dataset. This provides a total of 11,527 frames, and a total ground truth of 42,718 annotated TL bulbs. Additional information of the video sequences can be found in Table G.1. The resolution of the LISA Traffic Light Database is 1280x960, however only the upper 1280x580 pixels are used in this paper.

Table G.1: Overview of night test sequences from the LISA Traffic Light Dataset.

Sequence name	Description	# Frames	# Annotations	# TLs	Length
Night seq. 1	night, urban	4,993	18,984	25	5min 12s
Night seq. 2	night, urban	6,534	23,734	62	6min 48s
		11,527	42,718	87	12 min

[1] Freely available at `http://cvrr.ucsd.edu/LISA/datasets.html` for educational, research, and non-profit purposes.

5.2 Evaluation Criteria

In order to insure that the evaluation of TL detectors provide a comprehensive insight into the detectors performance, it is important to use descriptive and comparable evaluation criteria. The presented detectors are evaluated based upon the following four criteria:

PASCAL overlap criterion defines a true positive (TP) to be a detection with more than 50 % overlap over ground truth (GT).

Precision is defined in equation (G.1).

$$Precision = \frac{TP}{TP + FP} \tag{G.1}$$

Recall is defined in equation (G.2).

$$Recall = \frac{TP}{TP + FN} \tag{G.2}$$

Area-under-curve (AUC) for a precision-recall (PR) curve is used as a measure for the overall system performance. A high AUC indicates good performance, an AUC of 100% indicates perfect performance for the testset.

5.3 Results

We present the final results according to the original PASCAL overlap criteria of 50 % in Figure G.8 and G.9.

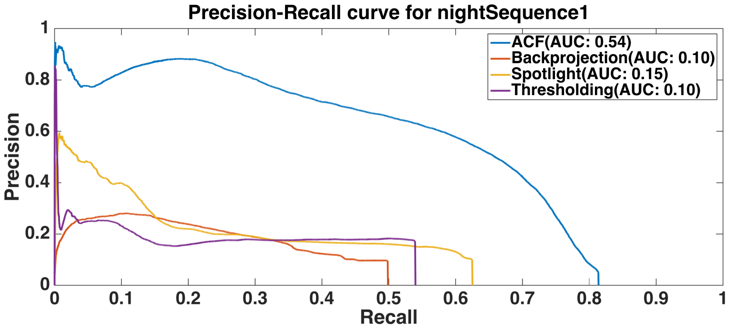


Fig. G.8: Precision-Recall curve of night sequence 1 using 50 % overlap criteria.

By examining Figure G.8 and G.9, it is clear that the learning-based detector outperforms the other detectors in both precision and recall on both

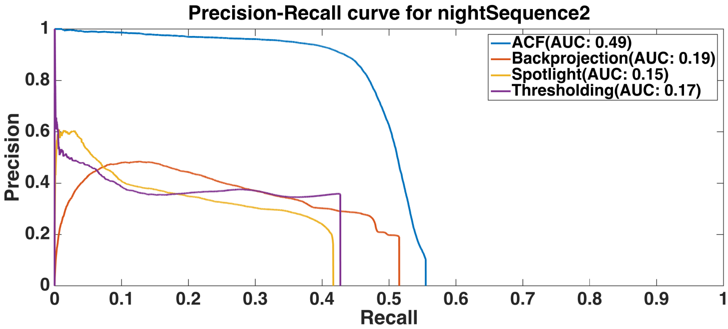


Fig. G.9: Precision-Recall curve of night sequence 2 using 50 % overlap criteria.

night sequences. The odd slopes of the PR curves for the back projection detectors are a result of problems with getting filled and representative BLOBs. The learning-based detector is able to differentiate well between TLs and other light sources, leading to a great precision and smooth precision-recall curve. The main problems with the learning-based detector's PR curves are the false negatives caused by detections not meeting the PASCAL criteria but still reaching a very high score, and problems with detecting TLs from far away. These detections causes some instability in the precision especially around 0.05 recall in Figure G.8.

6 CONCLUDING REMARKS

We have compared three detectors based on heuristic models to a learning-based detector based on aggregated channel features. The learning-based detector reached the best AUC because of the significantly higher precision and good recall. Recall is generally seen as the most important performance metric for detectors since precision can be improved in later stages, whereas false negatives are lost for good. The learning-based detector achieves an average AUC of 51.4 % for the two night test sequences. The heuristic model-based detectors achieved average AUCs ranging from 13.5 % to 15.0 %, with detection by back projection and spotlight detection achieving the highest AUCs.

Interesting future TLD work could involve applying and comparing the performance of deep learning methods on the LISA TL Dataset with the results presented in this paper.

REFERENCES

[1] Federal Highway Administration. (2009) Reducing late-night/early-morning intersection crashes by providing lighting. [Online]. Available: http://safety.fhwa.dot.gov/intersection/resources/casestudies/fhwasa09017/fhwasa09017.pdf

[2] M. B. Jensen, M. P. Philipsen, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, "Vision for looking at traffic lights: Issues, survey, and perspectives," in *Intelligent Transportation Systems, IEEE Transactions [In submission]*. IEEE, 2015.

[3] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," *18th IEEE Intelligent Transportation Systems Conference*, 2015.

[4] M. Diaz-Cabrera and P. Cerri, "Traffic light recognition during the night based on fuzzy logic clustering," in *Computer Aided Systems Theory-EUROCAST 2013*. Springer Berlin Heidelberg, 2013, pp. 93–100.

[5] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Proceedings of ICRA 2011*, 2011.

[6] C. Jang, C. Kim, D. Kim, M. Lee, and M. Sunwoo, "Multiple exposure images based traffic light recognition," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1313–1318.

[7] M. Diaz-Cabrera, P. Cerri, and P. Medici, "Robust real-time traffic light detection and distance estimation using a single camera," *Expert Systems with Applications*, pp. 3911–3923, 2014.

[8] H.-K. Kim, Y.-N. Shin, S.-g. Kuk, J. H. Park, and H.-Y. Jung, "Night-time traffic light detection based on svm with geometric moment features," *World Academy of Science, Engineering and Technology 76th*, pp. 571–574, 2013.

[9] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recogn.*, vol. 43, no. 1, pp. 416–430, Jan. 2010.

[10] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D Semantic Scene Priors for Online Traffic Light Interpretation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015.

[11] P. Dollár. (2015) Piotr's Computer Vision Matlab Toolbox (PMT). [Online]. Available: http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html

[12] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "Tracking both pose and status of a traffic light via an interacting multiple model filter," in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[13] R. Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 333–338.

[14] R. de Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 358–363.

[15] D. Nienhuser, M. Drescher, and J. Zollner, "Visual state estimation of traffic lights using hidden markov models," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1705–1710.

[16] Y. Zhang, J. Xue, G. Zhang, Y. Zhang, and N. Zheng, "A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles," in *33rd Chinese Control Conference (CCC)*, 2014, pp. 4924–4929.

[17] A. Mogelmose, D. Liu, and M. M. Trivedi, "Traffic sign detection for us roads: Remaining challenges and a case for tracking," in *Intelligent Transportation Systems*. IEEE, 2014, pp. 1394–1399.

[18] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features." in *BMVC*, vol. 2, 2009, p. 5.

Paper H

Improving a real-time object detector with compact temporal information

Martin Ahrnbom, Morten B. Jensen, Kalle Åström, Mikael Nilsson, Håkan Ardö, and Thomas B. Moeslund

The paper has been published in the
International Conference on Computer Vision Workshop (ICCVW): Computer Vision for Road Scene Understanding and Autonomous Driving workshop,
pp. 190–197, 2017.

The layout has been revised.

ABSTRACT

Neural networks designed for real-time object detection have recently improved significantly, but in practice, looking at only a single RGB image at the time may not be ideal. For example, when detecting objects in videos, a foreground detection algorithm can be used to obtain compact temporal data, which can be fed into a neural network alongside RGB images. We propose an approach for doing this, based on an existing object detector, that re-uses pretrained weights for the processing of RGB images. The neural network was tested on the VIRAT dataset with annotations for object detection, a problem this approach is well suited for. The accuracy was found to improve significantly (up to 66%), with a roughly 40% increase in computational time.

1 INTRODUCTION

Neural networks designed for real-time object detection using a single image as their input have recently improved significantly. Detectors like SSD [1], SqueezeDet [2] and YOLOv2 [3] outperform previous real-time detectors while approaching the accuracy of slower methods like those based on Faster R-CNN [4]. It might thus be tempting to use real-time detectors directly, but in practical problems there is often more information available than these networks take advantage of. For example, when detecting objects in videos, looking at only a single frame at the time is bound to make detection more difficult; humans have access to all we have seen before a given moment to help us detect various objects, and this information could be particularly helpful for occluded or small objects, hard to distinguish in a single frame. Commonly used datasets for object detection like COCO [5] and PASCAL VOC [6] only contain stand-alone images which algorithms are supposed to find objects in. This has led to strong development of algorithms and networks designed for this particular task. The use of temporal information in neural networks for object detection is not as well explored.

Taking advantage of temporal information in object detectors is not trivial. End-to-end learning is currently often the preferred way of solving computer vision and deep learning problems, but in the case of videos, that approach is not ideal. Feeding multiple frames directly into a Convolutional Neural Network (CNN) is problematic, as the amount of data to be processed by the network grows large if more than a few frames are to be considered. Recurrent Neural Networks (RNN's) can learn to process videos, but this only solves part of the problem; in order to properly train the RNN, it should be unrolled to allow backpropagation "through time" which also uses a large amount of memory during training if a large number of frames are to be

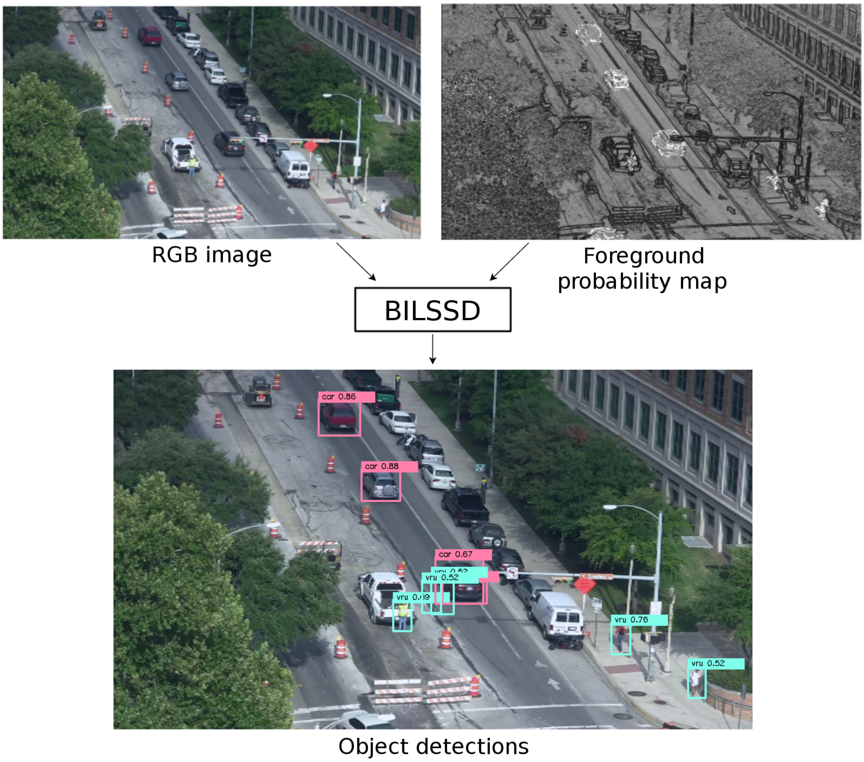


Fig. H.1: The BILSSD network takes an RGB image and a corresponding foreground probability map as input to produce object detections.

considered. If there was a compact way to represent temporal information gathered from a large number of frames, a faster and simpler approach would be to feed that data alongside standard RGB pixels into a single-frame object detector.

In the case where videos are filmed by a static camera, a foreground detector like the one by Ardö and Svärm [7] can be used to compute a per-pixel foreground probability map. Going from RGB to Red-Green-Blue-Foreground (RGBF) adds only a single input layer, increasing the amount of data to feed into the network only by 1/3 while providing useful temporal information. Compared to using an RNN, some generality is lost, as any temporal information other than what is considered foreground and background cannot be learned, and the videos have to be filmed by a static camera. What is gained is the simplicity and speed of being able to re-use existing and optimised single-frame object detectors as a starting point. Compared to using a single-frame object detector directly, temporal information is gained without sacrificing real-time performance.

Using foreground detection or background subtraction for object detection is a well-established concept, and used to be a popular approach for object detection. With the recent improvements to object detection CNN's, methods relying on foreground detection are no longer considered state of the art. However, this does not necessarily imply that these kinds of data cannot improve the performance of object detectors.

In other problems, other kinds of data might be available that can be expressed as an additional input layer. For example depth information, which has become more commonly available thanks to products like the Kinect, and thermal cameras that are sometimes used as a complement to RGB. The network design for including additional input layers does not need to make strict assumptions on the type of data it will process, as long as it somewhat resembles an image and is spatially correlated to the RGB layers.

For a network using RGB and additional modalities to be practically viable, unless a very large and varied annotated multimodal dataset is available for pretraining, it is necessary to be able to reuse RGB pretraining on the part of the network that is to process RGB data. It is also beneficial if the network can easily be constructed from any RGB-only object detector, such that if a better single-frame object detector is designed in the future, a corresponding improved mutlimodal version is easy to construct.

For practical object detection problems, before using a standard single-frame RGB object detector, one should ask if any additional data is available that could significantly help the detector perform its task, like temporal information. If so, a network design is needed that allows the use of this additional information, preferably without sacrificing the recent improvements of fast single-frame RGB object detectors. This paper proposes such a network.

The main contributions of this paper are:

- Bonus Input Layer Single-Shot multibox Detector (BILSSD), a novel neural network design based on SSD which can utilise both RGB and and additional data, like a foreground probability map, for object detection (Section 3)

- A set of annotations designed for object detection for some frames in the VIRAT [8] video dataset (Section 6.1)

2 RELATED WORK

Multimodal object detection has been attempted before. For example, Viola [9] and Jones and Snow [10] propose object detectors for videos using both spatial and temporal information that are not based on deep learning, distancing themselves from today's state-of-the-art approaches. Similarly, Gould [11] used RGB along with depth images for detecting household objects. Like BILSSD, features were calculated from both modalities separately, allowing some pretraining to be done on larger RGB-only datasets, but it was also not based on deep learning. Further from BILSSD's approach, Javed [12] and Bang [13] propose methods not based on deep learning using only temporal data (recurrent motion images and adaptive background subtraction images, respectively) for object detection.

One way of using temporal information for object detection is via recurrent neural networks. Ning [14] suggests a method for adding recurrent layers to an existing single-frame object detector to do simultaneous object detection and tracking. The high level features and detections from the single-frame object detector are fed into LSTMs that are trained to make spatially and temporally consistent detections. Because the recurrent layers operate on high level features, there is no ability to learn low-level motion features like separating the foreground from background. Such low-level features will not be brought up to high-level layers, as the single-frame object detector is first trained on its own, before the training of the recurrent layers.

Using temporal information for generating object proposals has been done in a few ways. Tripathi [15], Sharir [16] and Oneata [17] propose different methods for creating object proposals in videos, not only spatially but also temporally. Those object proposals can then be evaluated by a CNN to do full object detection in videos. These approaches differ significantly from BILSSD, as it does not rely on separate object proposals, which by necessity is computationally redundant as the tasks of finding and classifying objects are intimately connected.

Many neural networks use temporal information in videos for various other computer vision tasks. Yeung [18] propose a method for finding the times for certain actions in short videos by feeding multiple frames into an RNN. Karpathy [19] explore multiple ways of utilising temporal information for classifying entire videos, by comparing early, late and "slow" fusing strategies. The "slow" strategy fuses features in multiple steps, including some fusion in the middle of the network, somewhat similarly to BILSSD. Donahue [20] propose an RNN for image retrieval and caption generation in videos. Closer to BILSSD's approach, Simonyan [21] propose a neural network that process both RGB frames and optical flow differences between frames separately and classify videos by a late fusion of features from both modalities.

There have been deep neural networks that tackle the problem of foreground segmentation in videos, like Caelles [22]. It differs from traditional foreground segmentation algorithms in that it only segments a single foreground object, which has to be annotated manually in one frame. Another recent attempt at foreground segmentation is a neural network proposed by Jain [23], which does not utilise temporal information.

Gupta [24] train neural networks with depth data alone and in addition to RGB. They also take advantage of existing RGB networks by splitting the depth channel into three channels in an attempt to mimic the structure of RGB, and then retraining an existing RGB R-CNN detector on this new input data. This allows the re-use of an existing network design and pretrained weights, but they were not able to improve the results by fusing the modalities inside the network; instead they propose running two separate detectors and fusing their output.

In conclusion, many research approaches have tried to use temporal or otherwise multimodal input data for various vision tasks, including object detection, but none of them have made an object detector based on modern real-time neural networks, that combine RGB and temporal data in a "deep fusion" way, while being able to largely re-use the network design, and pretraining for the RGB processing layers.

3 BILSSD

This section describes a deep neural network design called Bonus Input Layer Single-Shot multibox Detector (BILSSD) based on Single-Shot multibox Detector [1]. The main difference is that BILSSD takes four input layers instead of three (RGBF instead of RGB, in our experiments). In order to be able to re-use initial layers pretrained for RGB images, the fourth input layer is processed separately by similar convolutional and pooling layers. The only difference in these layers is that the number of output features per layer is reduced by half, a design choice made on the assumption that the additional data can be represented by fewer features compared to RGB images. All features are then merged by three convolutional layers before being fed into the detection part of the SSD network. See Figure H.1 for a basic overview, and the top part of Figure H.2 for a more detailed description of the network. The design can be described as a "deep fusion", which differs from both "early fusion" and "late fusion" as the network processes the data both before and after the fusing of modalities.

Since the primary purpose of this network is to show the usefulness of providing additional input data, no other redesigns of the network, compared to standard SSD, are made.

BILSSD's concept of "deep fusion" is not inherently tied to SSD's design. Any similar deep neural network designed for object detection, for example YOLOv2 [3] and SqueezeDet [2], should be possible to modify in a similar way. The detection part of the SSD network is in BILSSD's implementation completely unchanged, and the processing of additional data is very similar to the processing of RGB, so making similar changes to any similar detector should be straightforward.

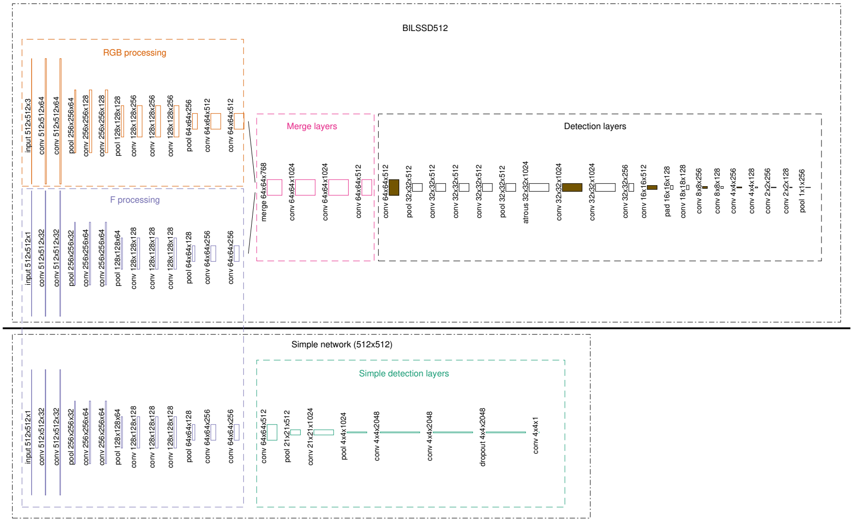


Fig. H.2: In this schematic, data flows from left to right. For each layer in the network, drawn as a box, the output dimensions are drawn on the left. Above horizontal black line is the network design of BILSSD512. RGB images are processed by the RGB processing layers (orange), and F are in parallel processed by the F processing layers (purple). Features from both are merged and processed by the merge layers (magenta). These are followed by the detection layers (black), which are identical to those in standard SSD. The boxes used as input into the SSD detectors (not shown in this visualisation) are filled brown. If one were to remove the F processing and merge parts, the result would be the standard SSD network. Note that the input and output of the merging layers are identical, meaning that no change to the detection layers was necessary. Below the horizontal black line is the "simple" network, which shares its initial layers with the F processing layers of BILSSD. The simple detection layers (green) do simplified object localisation and bring the resolution down to 4 × 4, which is the output of the simple network.

BILSSD's design is not inherently bound to some specific type of additional data; as long as the data can be expressed as a single layered image that is spatially consistent with the RGB data, it can be used with BILSSD, although minor changes like the number of output features from the layers processing the additional data may improve results, depending on the type of data.

4 PRETRAINING FOREGROUND FEATURE EXTRACTION

While the first few layers that process RGB images based on VGG-16 can utilise existing pretrained weights to initialise the training process, no corresponding weights exist for the layers that process the foreground probability maps. It was initially tested to train the BILSSD network with pretraining for the RGB layers, and randomly initialised weights for the others layers. The network was then found to prefer only using RGB features. To work around this, a simple neural network was designed, which shares the initial layers with BILSSD's initial layers that extract features from foreground probability maps. The task of this simple network is to, given a foreground probability map, produce a 4 × 4 grid of values between 0 and 1, where high values indicate high confidence that an annotated object (of any class) exist in the corresponding 16th of the image, and low values indicate the opposite. An example of what output from the simple network can look like can be seen in Figure H.3.

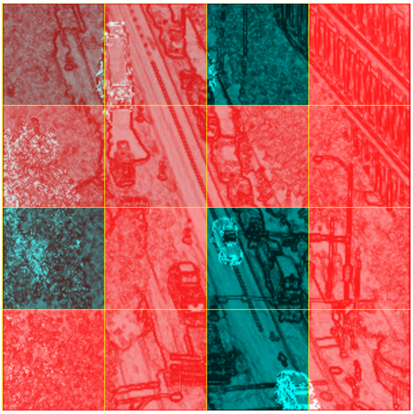


Fig. H.3: An example of output from the simple network. Blue regions (dark regions, if viewed in monochrome) means high confidence for annotated objects appearing somewhere in the region, while red (brighter, if viewed in monochrome) means a low confidence. These colours are drawn over the 300 × 300 foreground probability map that the simple network receives as input. The image is rescaled to this size before being processed by the foreground detection algorithm. In this example, the simple network is able to correctly detect two cars and a pedestrian, but also believes the moving tree to be an annotated object.

Converting existing ground truth to this format is straight-forward, by marking the cell in the 4 × 4 grid containing the center coordinates of each annotation's bounding box as a 1, while all others are set to 0. The simple network is designed to learn to find objects rather than to classify them. The idea behind this design is that the foreground probability maps may be better suited for localisation rather than classification.

After training this simple network, the weights for the layers that overlap with the processing of the additional input data in BILSSD are used as pretrained weights. This approach should help BILSSD utilise the additional input data. For a detailed description of the simple network for processing 512×512 images, see the bottom part of Figure H.2. When processing 300×300 images, the only difference to when processing 512×512 images is the last pooling layer which pools a 3×3 region rather than 5×5, to bring the resolution down to the same 4×4.

5 BGGRAD FOREGROUND DETECTION

The foreground detection algorithm used in this paper is BGGRAD, as described in Ardö [7]. This algorithm generates a single-layer probability map where dark pixels indicate a high probability of background, white pixels indicate a high probability of foreground and grey areas are regions where the algorithm is not certain. Because the algorithm is based on matching gradients directions in different frames, areas with little or no gradients, like flat surfaces (generally in the interior of objects) will appear grey. This means that, in general, foreground objects appear grey with white outlines while background objects appear grey with black outlines. A few examples can be seen in Figure H.4. This allows the shapes of objects to remain visible, and could help the network in separating the different objects when looking at only the foreground probability maps.

The algorithm's main limitation, like most foreground detection methods, is that it relies on the camera being stationary. When the camera shakes, background objects will appear as foreground. It is also somewhat sensitive to heavy compression artifacts, as edges between blocks of pixels compressed separately may appear as foreground.

6 EXPERIMENTS ON VIRAT

A Keras implementation[1] of BILSSD was trained on the VIRAT dataset using annotations designed for object detection (see Section 6.1). The output from the BGGRAD foreground detection algorithm [7] was used as the fourth input layer. For this task, BILSSD was trained and evaluated using RGBF, only RGB and only F. This allows some analysis of how much the different modalities help in object detections. When only using one modality, this was im-

[1] The implementation is based on a port of SSD to Keras available here: `https://github.com/rykov8/ssd_keras`

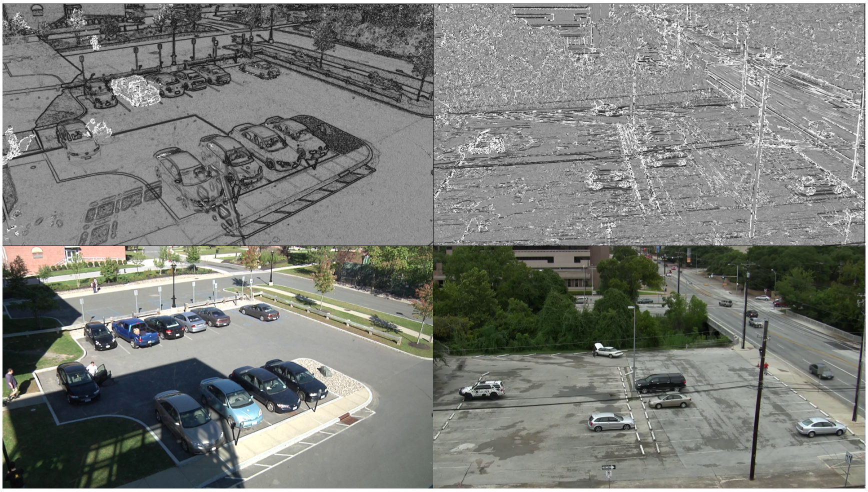


Fig. H.4: Two examples of the BGGRAD algorithm after running on videos from the VIRAT dataset. At the bottom are the RGB inputs, and above them are the corresponding foreground probability maps. On the right, is an example of what the output looks like when the algorithm runs on a shaky video, where all edges appears as foreground.

plemented by feeding only zeroes as input to the other modality's processing layers. In the case where only RGB is used, BILSSD should behave nearly identical to the standard SSD network in terms of accuracy, as the only difference is the additional "merging" layers that are assumed to affect the end result at most marginally, as they should quickly learn to only include RGB features.

In these experiments, both the 300 × 300 and 512 × 512 versions of SSD were used as the base for BILSSD, and the two versions are labelled "BILSSD300" and "BILSSD512". Images were scaled down to 300 × 300 and 512 × 512 respectively before going through the background detection algorithm. To generate the foreground probability maps, all frames in the videos were fed into the foreground detection algorithm. The frames where annotations exist were saved, and used in training.

6.1 VIRAT annotations for object detection

The VIRAT dataset [8] is designed for event recognition, and thus its official annotations only mark certain pedestrians and vehicles that are part of the annotated events. The dataset is however a large collection of surveillance videos filmed with, for the most part, stable cameras, making it a good benchmark for object detections that work in such a context. We have made third-party annotations for the task of object detection, where most visible objects of the following two classes are annotated by bounding boxes:

- "vulnerable road users" ("VRU's", such as pedestrians, bicyclists)
- "vehicles" (four-wheeled vehicles like cars, buses, trucks)

A total of 1240 frames have been annotated, from 62 different videos in the VIRAT dataset. Half of those videos make up the training set, while the other half is used for evaluation. In total, there are 2733 VRU's and 721 vehicles annotated, with 1368 VRU's and 339 vehicles in the training set, and 1365 VRU's and 382 vehicles in the test set. The annotations are made to resemble data used in traffic surveillance analysis, for example only vehicles that are not parked are annotated. This, along with a large number of small pedestrians that likely appear more clearly in foreground probability maps, makes utilising temporal information a promising approach for this challenge. On the other hand, there are frames in the dataset where camera shake cause the foreground detection algorithm to produce bad results. These annotations are available here: `https://github.com/ahrnbom/ViratAnnotationObjectDetection`.

6.2 Training

First, the simple network was trained on foreground probability maps using randomly initialised weights, for 30 epochs. This procedure was repeated until a good initialisation allowed convergence. The network was tested with these weights on some samples from the dataset and its output was inspected visually to make sure the simple network had learnt to detect objects. This was done for both 300×300 and 512×512 foreground probability maps.

For all three variants (RGBF, RGB and F) the BILSSD300 and BILSSD512 networks were trained for 100 epochs using an Adam optimiser [25] with a base learning rate of 3×10^{-4}. The batch size was 16 for BILSSD300 and 8 for BILSSD512. The BGGRAD foreground detection algorithm was set to look at the previous 100 frames for computing the foreground probabilities, and it processes blocks of 8×8 pixels at the time.

For the RGB processing layers, pretrained weights from ImageNet [26] were used, while the F processing layers used weights from the simple network as described in Section 4. The merging and detection layers had random weight initialisation.

During training, data augmentation was performed by horizontally flipping the images with a probability 0.5, varying saturation, brightness, contrast and lighting over RGB images, while adding random noise with an amplitude of 10% of the value range to the foreground probability maps. Additionally, random cropping was performed with an aspect ratio between 3/4 and 4/3 and the area of the cropped section was between 75% and 100% of the original images.

All variants (RGBF, RGB and F) were trained for 100 epochs each, which took around 3 hours for SSD300 and 6 hours for SSD512.

6.3 Results

Accuracy

The mAP scores for BILSSD300 and BILSSD512 for the VIRAT dataset can be seen in Table H.1, and corresponding precision-recall curves for the two classes can be seen in Figure H.5. In short, using RGBF outperforms using only RGB (which should behave similarly to standard SSD) or only F in terms of accuracy. The accuracy improves for both the tested input resolutions, by 66% and 31% respectively.

	RGBF	RGB	F
BILSSD300	0.272	0.208	0.154
BILSSD512	**0.400**	0.241	0.125

Table H.1: mAP scores for different input resolutions and modalities of BILSSD on the VIRAT dataset. Bold number indicates best result.

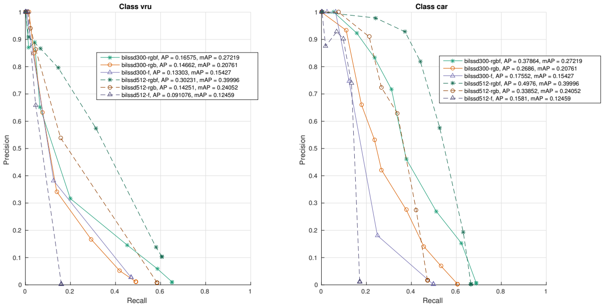


Fig. H.5: Precision-recall curves for the VIRAT dataset. Using RGBF is better than using only RGB or only F for both resolutions, and the improvement is significant in all cases except for the VRU class in lower resolution, where the improvement is marginal. Higher resolution is better than lower resolution for RGBF and RGB, but surprisingly not for only F, which performs poorly overall.

Qualitative analysis

Some output from the different BILSSD networks trained on RGBF, RGB and F were inspected manually. It was found that when trained with only F or only RGB, correct detections were given only marginally better confidence values than a large number of incorrect detections. They all had problems with giving false positives relatively high confidences, around 0.40 for RGBF and around 0.44 for F and RGB, while true detections vary between 0.4 and 0.9 for RGBF but for F and RGB they only rarely get above 0.5. Using only RGB, confidences above 0.5 were more common than when using only F, explaining the low mAP scores of the latter. True positives of the VRU class generally got lower confidences than of the vehicle class, likely due to the smaller objects being harder to detect.

Comparing 300×300 and 512×512 versions of RGBF, the higher resolution was found to help detecting small objects. They both had problems with outputting more than one box near real objects, not quite close enough to be caught by the non-maximum suppression, and this issue is more noticeable in the lower resolution. Failing to localise objects did occur, as well as some false positives, but incorrect classifications were uncommon.

Execution speed

For the computer used in these experiments, which is equipped with a NVIDIA Titan X GPU and an Intel Core i7-6800K CPU, the execution times for a batch size of 1 can be seen in Table H.2. These times can be compared to the original SSD's reported execution speeds on the same GPU model and batch size, which were 46 FPS and 19 FPS for SSD300 and SSD512 respectively. One should note that the original SSD implementation was done in Caffe, while BILSSD's implementation was done in Keras with a Tensorflow backend and the computers' other differences may have some impact, so the numbers are not perfectly comparable. However, using these numbers, BILSSD (including the BGGRAD preprocessing) has roughly 40% more processing time compared to SSD.

7 CONCLUSIONS

We have introduced the BILSSD network, which is based on SSD while adding the ability to utilize multimodal spatially aligned input. We have tested it on the VIRAT dataset, using foreground probability maps computed by the fast BGGRAD foreground detection algorithm as the additional input, along with RGB images.

On this dataset, accuracy increases when RGB and F are used together, compared to using only RGB and using only F, for both the VRU and vehicle classes. The improvements are expected, as it is difficult to tell the difference between a parked and non-parked car without temporal data, and small pedestrians appear more clearly in the foreground probability maps.

	BILSSD300	BILSSD512
Time BILSSD	0.029 s	0.055 s
Time BGGRAD	0.0023 s	0.019 s
Time both	0.031 s	0.074 s
FPS BILSSD	34 FPS	18 FPS
FPS BGGRAD	430 FPS	52 FPS
FPS both	32 FPS	14 FPS

Table H.2: Execution times and frame rates for BILSSD and BGGRAD on 300 × 300 and 512 × 512 resolutions. These times were computed as an average over more than 100 frames.

For example, BILSSD300 using RGBF outperforms BILSSD512 using only RGB (similar to SSD512) in terms of mAP while running much faster, so for this problem, adding temporal data is a more efficient way to improve performance than increasing the resolution. Accuracies for the VRU class are generally low for BILSSD300, which makes sense as most instances of this class are small, making them more difficult to detect in low resolution images.

Using only F performs poorly overall. Because the features learned from the F input improves RGBF significantly compared to RGB, it is obvious that these features are helpful, but on their own they do not seem to provide enough confidence for separating true from false positives. Another reason why using only F performs so poorly could be the lack of training data; while the RGB layers use the large and varied ImageNet as a starting point, the F layers have only ever looked at the limited number of images in the VIRAT annotations. Finding a better pretraining strategy for the F layers could improve not only the accuracy when using only F, but also when using RGBF.

It should be noted that the relatively low number of annotated frames used in these experiments means that one should be careful drawing too general conclusions about how much using foreground probability maps in addition to RGB improves accuracy on other datasets. What can be concluded is that the BILSSD network is capable of utilising multiple modalities to increase the accuracy of object detections, and this improvement is independent of increasing the spatial resolution of the input data.

8 FUTURE WORK

There are other ways in which BILSSD could be evaluated. Most notably, it would be interesting to try the approach on more datasets. There is currently no large scale dataset of videos filmed with stationary cameras in varied environments in decent video quality. The DETRAC dataset [27] is a large scale surveillance dataset, but the camera shake in most videos prevent the BGGRAD algorithm to work properly. Developing a fast yet shake resistant foreground detection algorithm and using it with BILSSD on the DETRAC dataset could be an interesting direction for future evaluation.

The pretraining of the foreground processing layers could likely be improved. As Gupta [24] showed, pretraining on RGB images from ImageNet can be used as an initialisation for non-RGB images with improved results compared to starting from scratch, if the data is formatted to partially mimic the structure of RGB. Perhaps a foreground detection algorithm could be developed which produces three output layers, somewhat mimicking the structure of RGB, to take full advantage of such an approach. The probability map already shares some properties with RGB, like the concept of edges around objects, so such an approach is probably feasible.

It would be interesting to try the BILSSD network using other modalities than foreground probabilities alongside RGB, like depth data or thermal images, perhaps using more than one non-RGB modality at the time. Different pretraining strategies and minor network changes may be necessary, depending on the data.

Implementing similar multimodal versions of other object detectors, like Faster R-CNN, SqueezeDet and YOLOv2, would allow further analysis of how well the concept of deep fusion generalises. When a new and better single-frame object detector is made in the future, as long as this network can be split into a feature extraction part and a detection part, implementing a deep fusion of modalities in the style of BILSSD should be easy.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, “SSD: single shot multibox detector,” *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[2] B. Wu, F. N. Iandola, P. H. Jin, and K. Keutzer, “Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving,” *CoRR*, vol. abs/1612.01051, 2016. [Online]. Available: http://arxiv.org/abs/1612.01051

[3] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[5] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[7] H. Ardö and L. Svärm, "Bayesian formulation of gradient orientation matching," in *Lecture Notes in Computer Science*, vol. 9163. Springer, 2015, pp. 91–103. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-20904-3_9

[8] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*. IEEE, 2011, pp. 3153–3160.

[9] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vision*, vol. 63, no. 2, pp. 153–161, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-6644-8

[10] M. J. Jones and D. Snow, "Pedestrian detection using boosted features over many frames," in *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, 2008, pp. 1–4. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2008.4761703

[11] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, "Integrating Visual and Range Data for Robotic Object Detection," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008*. Marseille, France: Andrea Cavallaro and Hamid Aghajan, Oct. 2008. [Online]. Available: https://hal.inria.fr/inria-00326789

[12] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ser. ECCV '02. London, UK, UK: Springer-Verlag, 2002, pp. 343–357. [Online]. Available: http://dl.acm.org/citation.cfm?id=645318.649249

[13] J. Bang, D. Kim, and H. Eom, "Motion object and regional detection method using block-based background difference video frames," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on*. IEEE, 2012, pp. 350–357.

[14] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, and H. Wang, "Spatially supervised recurrent convolutional neural networks for visual object tracking," *arXiv preprint arXiv:1607.05781*, 2016.

[15] S. Tripathi, S. J. Belongie, Y. Hwang, and T. Q. Nguyen, "Detecting temporally consistent objects in videos through object class label propagation," *CoRR*, vol. abs/1601.05447, 2016. [Online]. Available: http://arxiv.org/abs/1601.05447

[16] G. Sharir and T. Tuytelaars, "Video object proposals," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 9–14.

[17] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *European conference on computer vision*. Springer, 2014, pp. 737–752.

[18] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[20] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc.,

2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[22] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool, "One-shot video object segmentation," *CoRR*, vol. abs/1611.05198, 2016. [Online]. Available: http://arxiv.org/abs/1611.05198

[23] S. D. Jain, B. Xiong, and K. Grauman, "Pixel objectness," *arXiv preprint arXiv:1701.05349*, 2017.

[24] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 345–360.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[27] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "DETRAC: A new benchmark and protocol for multi-object detection and tracking," *arXiv CoRR*, vol. abs/1511.04136, 2015.

Part IV

Semantic

Paper I

The RUBA Watchdog Video Analysis Tool

Chris H. Bahnsen, Tanja K. O. Madsen, Morten B. Jensen, Harry Lahrmann, and Thomas B Moeslund

This article has been submitted as a public deliverable within the *InDeV EU project*, 2018, available at https://www.indev-project.eu/InDeV/EN/Documents/pdf/4-3.pdf?__blob=publicationFile&v=2. The deliverable is based on the public wiki page at https://bitbucket.org/aauvap/ruba/wiki/.

Fig. I.1: The RUBA logo.

1 INTRODUCTION

The Road User Behaviour Analysis (RUBA) project is a watch-dog tool for computer-based analysis of traffic videos. The program can be used on Windows, MacOS, and Linux computers.

RUBA is developed by the Visual Analysis of People Lab at Aalborg University, Denmark, in collaboration with the Traffic Safety Research Group at Aalborg University.

RUBA allows the user to draw fields (detectors) on the video image by using a simple click-based drawing tool. The sensitivity of the detector, regarding movement in the image, is adjusted by different parameters in the program.

How to contribute

Please feel free to use RUBA and see if it fits your use case and research needs. If you encounter a bug by doing so, or if you have any suggestions on the further improvement of RUBA, please report it in our issue tracker.

License

RUBA is licensed under the MIT License.

2 ANALYSIS IN RUBA

The procedure when conducting an analysis in RUBA is as follows:

1. Import video(s)
2. Create module(s) for the analysis
3. Calibrate parameters to ensure that the right movements/road users are registered
4. Run the analysis

2.1 Import of videos

RUBA handles videos of most file types and resolutions. The program offers two different approaches for handling the synchronisation and time management for every frame of a video file:

1. If the frame rate of the video is constant, the start time of the video might be encoded into the file name of the video. The exact time of a frame will be computed based on the start time, the frame rate of the video, and the current frame number.
2. If the frame rate of the video is varying, you may put the exact date and time of each frame in a separate log file. The log file should be placed in the same directory as the corresponding video file and share the same file name except for the extension. As default, RUBA looks for corresponding files with the '.log' extension.

For more information on the video synchronisation options, refer to Section 4.2. Once you have selected a suitable way to ensure the synchronisation of the video, you have two options to import video files into RUBA, illustrated in Figure I.2 and listed below:

1. Use File -> Load Video Files or click the button at the menu bar (CTRL + O). This option will clear the current list of video files and import the new files that you have selected.
2. Use the 'Add videos to list (CTRL + INS)' button in the 'Video files' pane. This option will add the selected videos files to the bottom of the current list of video files.

2.2 Creation of modules for the analysis

After the videos have been imported the first video is shown in the window pane. A module for the analysis is created by pressing the button for the desired module. This is illustrated in Figure I.3.

different detectors is given in Section 5.

2.3 Drawing the mask

After the desired detector have been chosen a new window opens, shown in Figure I.5. This window contains the settings of the detector and lets the user draw the detector. Via `Configure detectors` the detector is chosen, after which drawing tools to create the detector and a number of detector settings appears. The settings depend on the chosen detector type.

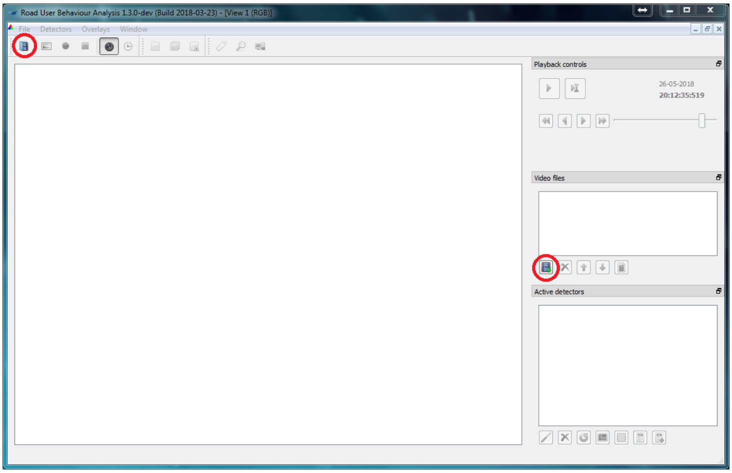


Fig. I.2: Import of videos is done via either of the two buttons marked in red.

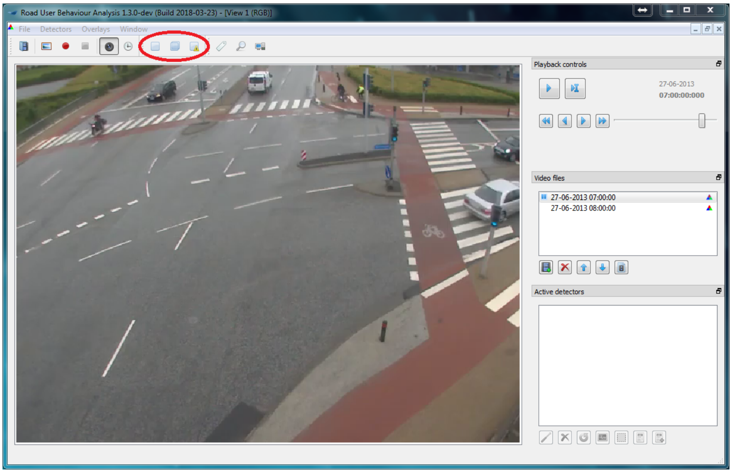


Fig. I.3: Creation of modules

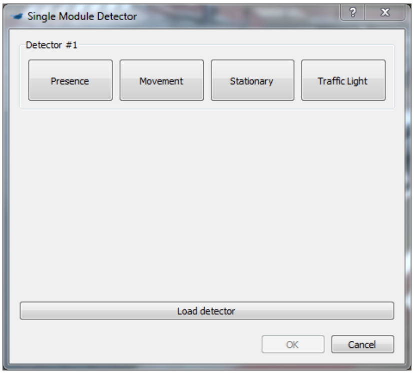
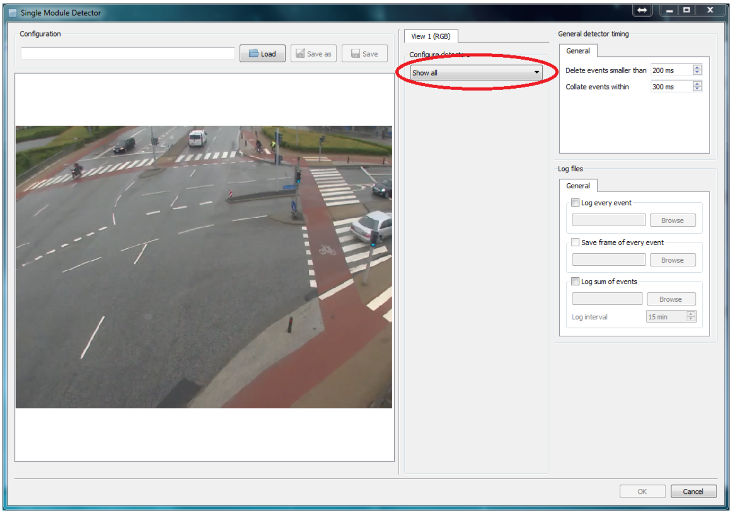


Fig. I.4: Choice of detector type in single module



Configure detectors: Show all

Delete events smaller than 200 ms

Collate events within 300 ms

□ Log every event

□ Save frame of every event

□ Log sum of events

Log interval 15 min

Fig. I.5: Creation of detectors.

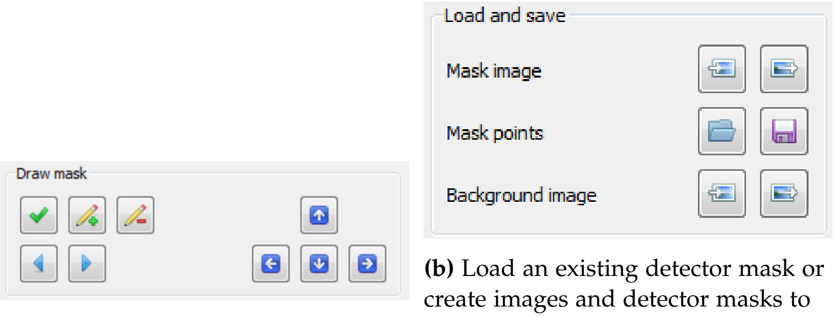
Table I.1: Functionality of the drawing tools in RUBA.

Button	Description
✎−	Remove the last corner point.
✎+	Add a point between the current and the previous corner.
◀ ▶	Switch between corners. The current point is marked with a circle.
↑ ← ↓ →	Move the corner up/down/left/right.

To draw the outline of the detector, click on the pencil. The detector is drawn by clicking in the image. Straight lines are created between the points. The latest point can be deleted by right clicking.

Drawing tips

Use the drawing tools illustrated in Figure I.6a to modify your detector. The functionality of the tools is explained in Table I.1.



(a) Tools for drawing the mask

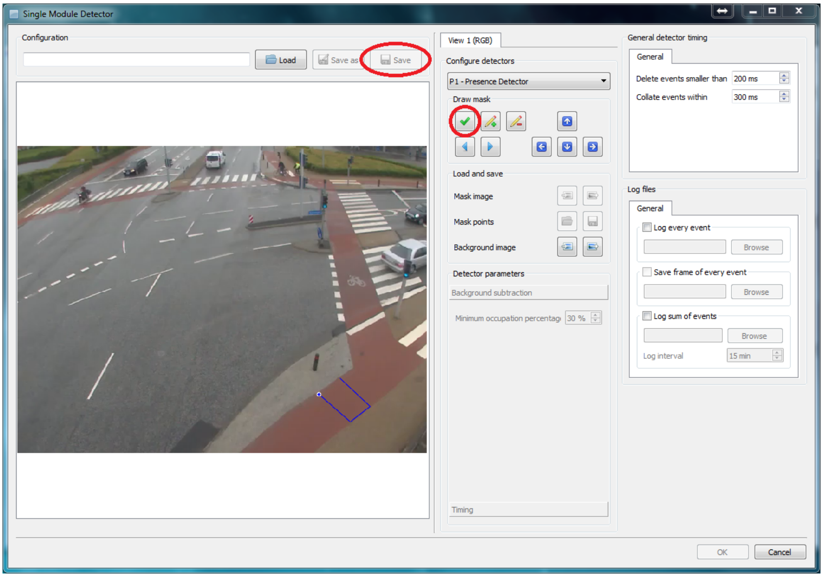
(b) Load an existing detector mask or create images and detector masks to reuse the masks

Fig. I.6: Mask manipulation tools in RUBA.

the buttons to find the keyboard shortcut.

Move points: When drawing your detector, you can click on the points and drag them to where you want them to be.

If you have an existing detector mask or want to save the mask area as an image or RUBA file, you can use the functionalities in the `Load and save` panel shown in Figure I.6b.



P1 - Presence Detector

Minimum occupation percentag 30 %

Delete events smaller than 200 ms

Collate events within 300 ms

☐ Log every event

☐ Save frame of every event

☐ Log sum of events

Log interval 15 min

Fig. I.7: Creation of detector using the drawing tool

- `Mask image` loads (left) or saves (right) an image of the detector where the detector is white and the background is black.
- `Mask points` can be used to save a RUBA configuration file with information on the size and position of the detector.
- `Background image` imports and exports a screen shot of the background.

Once the detector has been drawn (i.e. it only needs to be closed), double click or press the green tick mark, marked in red on Figure I.7. After this, the parameters can be adjusted, and it can be chosen if logs should be created. See Section 7.3 for detailed information on the log system of RUBA.

The detector is saved via the `Save`-button before the window is closed via the `OK`-button. Configuration files that have previously been saved can similarly be imported in this window.

2.4 Calibration of parameters

To ensure that the right objects are detected the parameters must be calibrated. This is done via a number of tools, marked in red in Figure I.8, which let the user gain insight into what is detected by the algorithms.

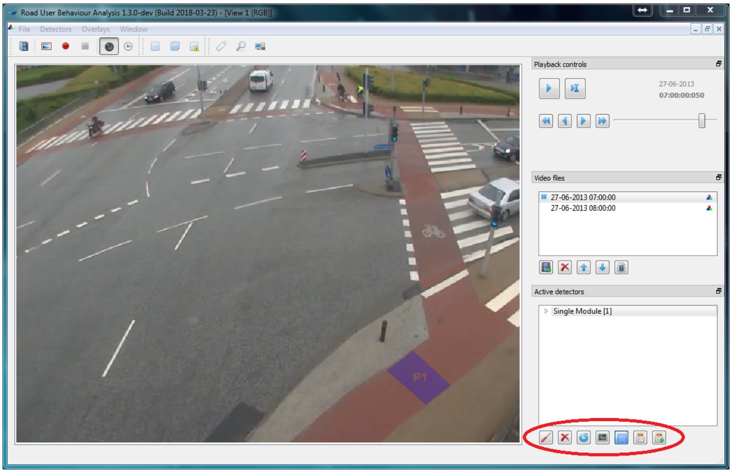


Fig. I.8: Tools for calibration of the detectors. From left: 1) edit the detector. Editing can also be done by right clicking on the detector in the `Active detectors`; window. 2) delete the marked detector. If the detector has not been saved, it is deleted completely and cannot be imported. 3) histograms. 4) Overlay of the detector in the image. 5) information about activity in the detector. 6) extended information about the detector.

2.5 Histograms

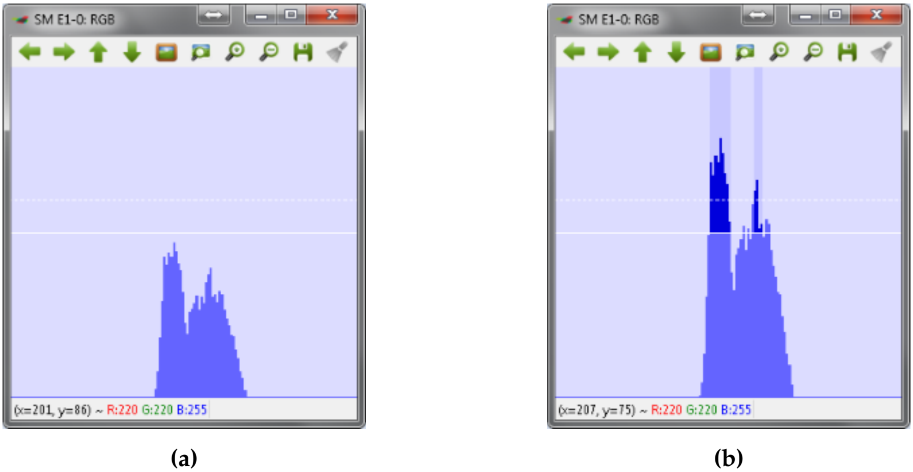
The most important tool for the calibration is the histograms of the activity in each detector.

Histograms are used to adjust the detectors. If the amount of activity is sufficiently high so that the software recognizes it as a road user, the activity is marked with a bright colour, illustrated in Figure I.9. If the parameters of the detector are not adjusted correctly, then the road user will either be missed or only partly detected, as seen in Figure I.9a. After the adjustments of the parameters the road user will be clearly detected.

The histograms of the movement detector, the stationary detector, and the traffic light detector are described in more details in Section 5.

When adjusting the detectors, change a few parameters at a time and validate experimentally if the change has any effect. The most important parameters of the detectors are listed below:

- **Presence detector:** Minimum occupation percentage
- **Movement detector:** Trigger threshold, movement range, and minimum speed
- **Stationary detector:** Minimum occupation percentage, minimum speed, and max vector count



(x=201, y=86) ~ R:220 G:220 B:255

(x=207, y=75) ~ R:220 G:220 B:255

(a) **(b)**

Fig. I.9: Calibrating the trigger threshold of the presence detector.

- **Traffic light detector:** The position of the annotated traffic light positions

A detailed description of all detector parameters is given in Section 5. Detailed information on how to set up the logger is given in Section 7.3. There, you will also find information on the timing options of the different detector modules.

2.6 Run the analysis

When the detectors have been calibrated the analysis can be performed. If it has not yet been specified which log files should be created during the analysis, this is done by double clicking the detector in *active detectors*. Run the analysis by pressing the `play` button which is marked in red on Figure I.10.

2.7 Inspecting the log files

If `log every event` or `log sum of events` have been marked when configuring the detector, a number of log files (.csv-files) will be generated. The log files may be inspected by a text editor or a spreadsheet program such as Microsoft Excel. More details on the log system are provided in Section 7.3.

2.8 Multi-threaded processing

It takes time to process long videos in RUBA, especially if the resolution of the video frames is high. In order to help with this problem, RUBA has

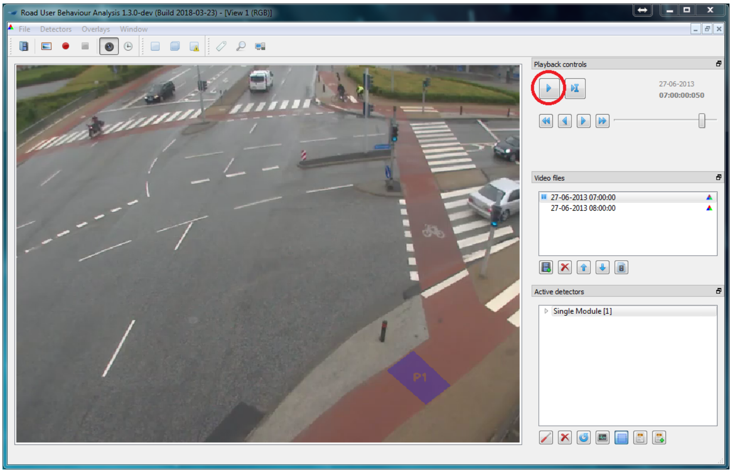


Fig. I.10: Running a traffic analysis. Double click on the first video to scroll back to the beginning of the video, and then press the play button to run the analysis. The analysis is complete when the time stops in the end of the last video. Please note, that the time and date must be specified in the beginning of each video if the user defined time format is used.

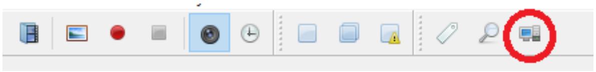


Fig. I.11: Opening the Multi-threaded processing dialogue.

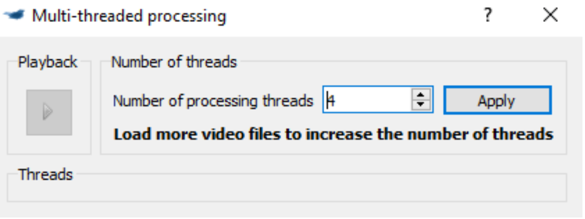
an option to split the analysis such that it runs on multiple threads. Once the videos are loaded and the detectors are initialised, press the `Perform multi-threaded processing` button in the main menu, marked in red in Figure I.11.

Once you have opened the multi-threaded processing dialogue, you may select the desired number of threads to perform the analysis. The maximum number of threads is dependent on the number of physical CPU-cores on your machine. Furthermore, in order to create a number of threads, each tread needs at least one unique video.

In the example in Figure I.12, RUBA has detected that the machine has eight CPU-cores, but RUBA only allows the creation of four threads because only four video files are loaded. In order to increase the number of threads, more video files should be provided and the multi-threaded processing dialogue should be reopened.

The optimal number of processing threads

The maximum number of threads is computed as the number of CPU-cores, minus 1. If the computer has four CPU-cores, three will be selected for run-



Number of processing threads 4

Load more video files to increase the number of threads

Fig. I.12: The initial window in the Multi-threaded processing dialogue.

ning the analysis - and the last will be spared for showing the progress in RUBA and for running other tasks.

As a general rule, the processing speed is proportional to the number of processing threads. However, if your CPU features hyper-threading technology, RUBA will typically see one physical CPU-core as two CPU-cores. On a machine that has four physical CPU-cores with hyper-threading, RUBA will report the maximum number of threads to `4 * 2 - 1 = 7`, seven threads. In this case, the addition of more threads than physical CPU-cores will have little impact on performance.

Running the multi-threaded analysis

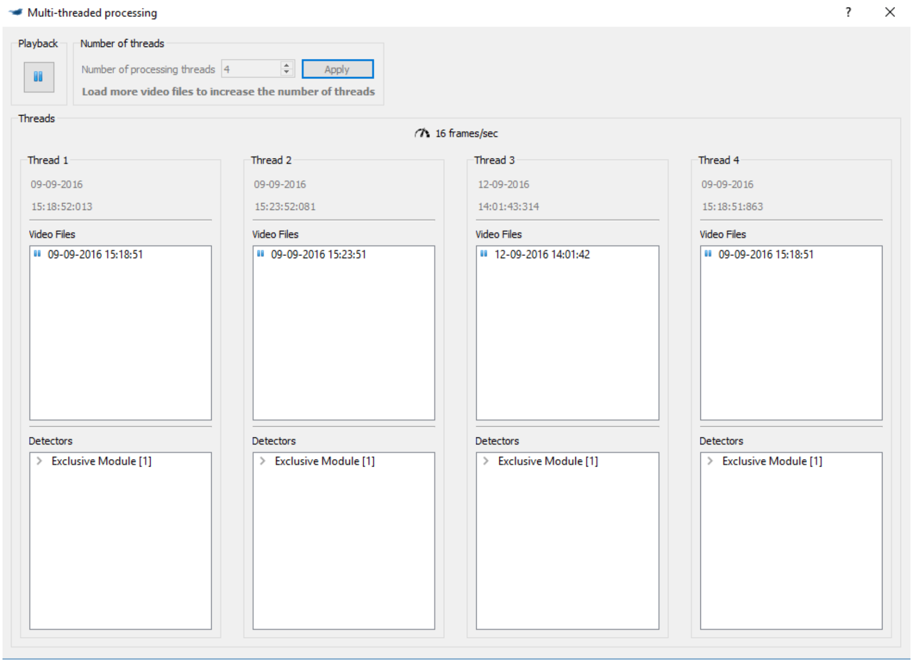
Once you have selected the desired number of processing threads, press the `Apply` button. Behind the scenes, RUBA will save the detectors and reload them for every thread. This might take some seconds depending on the number of threads and the size of the detectors. After this process has finished, the window will be resized and the desired number of threads are shown. A sample screen with four threads is shown in Figure I.13. Press the `Play` button in the upper left corner to start processing.

The `Video Files` window shows the progress of the analysis. As opposed to normal analysis, it is not possible to jump to a specific video by double-clicking.

The `Detectors` window allows you to inspect the progress by expanding the arrows, similar to a normal analysis. However, the following features are not supported when the multi-processing window is opened:

1. Reconfiguring detectors
2. Showing the detector masks
3. Showing the detector histograms
4. Overlaying debug information on the videos

Because the analysis now runs in parallel, you will find that RUBA creates temporary log files, one for each thread. Once all the threads has finished



Multi-threaded processing

Playback

Number of threads

Number of processing threads 4

Apply

Load more video files to increase the number of threads

Threads

16 frames/sec

Thread 1
09-09-2016
15:18:52:013
Video Files
09-09-2016 15:18:51
Detectors
Exclusive Module [1]

Thread 2
09-09-2016
15:23:52:081
Video Files
09-09-2016 15:23:51
Detectors
Exclusive Module [1]

Thread 3
12-09-2016
14:01:43:314
Video Files
12-09-2016 14:01:42
Detectors
Exclusive Module [1]

Thread 4
09-09-2016
15:18:51:863
Video Files
09-09-2016 15:18:51
Detectors
Exclusive Module [1]

Fig. I.13: The multi-threaded processing is started.

processing, RUBA will automatically combine the temporary log files into a single log file.

3 USER INTERFACE

Figure I.14 illustrates the main window of RUBA, after a video has been imported. Until then, most buttons are inactive. In the following, the function of each button is described. Keyboard short cuts are defined in square brackets.

1. Main menu. In the tabs, the same functions that are accessible via buttons in the main window can be found, as well as a number of program settings to use before conducting the analysis.

2. Import video(s) [Ctrl + O].

3. Take a screenshot of the video pane (20) [F9].

4. Record a video of the video pane. Clicking the red dot [F10] starts the recording. It is possible to pause the recording by clicking the button again. The recording can be resumed by clicking the red button again. When clicking the square button [ctrl + F10] the recording is finished.

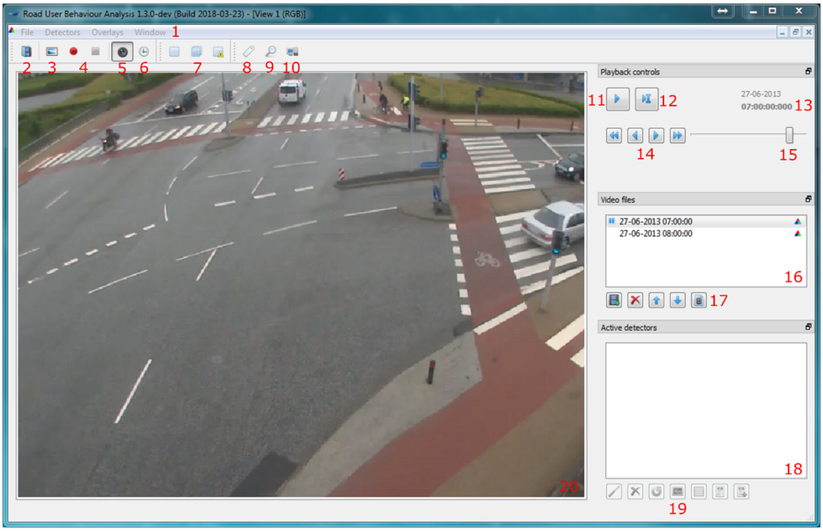


Fig. I.14: User interface shown when one or several videos have been imported.

All overlays (detectors, etc.) which are shown in the video pane (20) will appear in the recording.

5. Enable/disable that the video is shown in the video pane (20) [F3].

6. Add/remove overlay which shows a timestamp (13) in the video [F4].

7. Flexible analysis tool (detector) consisting of a single module (left) [Ctrl+1], a double module (middle) [Ctrl+2] and an exclusive module (right) [ctrl+2].

8. Annotate Ground Truth. Opens the `Ground Truth Annotator` panel which can be used to manually detect activity. These detections can be used to calibrate detectors.

9. Review Log Files. Opens the `Log File Reviewer` panel which can be used to review events from a log file and create a new log file with selected events.

10. Multi-Threaded Processing. Press this button to open the `Multi-Threaded Processing` panel and analyse the videos in multiple thread simultaneously to speed up the analysis.

11. Start/pause analysis [space].

12. Jump to a specific frame in the video.

13. Date and time for the video.

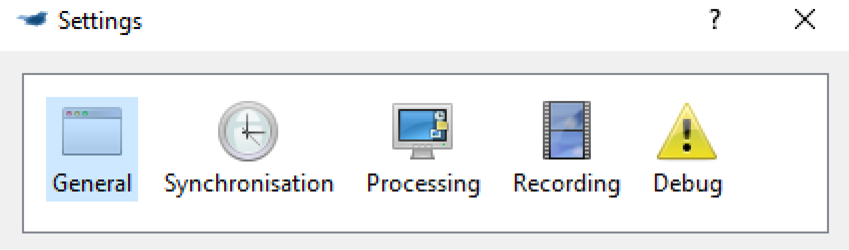


Fig. I.15: The settings panel in RUBA. The `Debug` panel is only shown when running a development version of RUBA.

14. Navigate through the video. Use these four buttons to respectively jump five frames previous [A], one frame previous [S], one frame forward [D] and five frames forward [F].

15. Adjust video speed. When the slider is placed to the far left the video is paused. When the slider is placed to the far right the video is sped to the maximum.

16. Imported videos. The video that is currently played is marked with a *pause* symbol.

17. Respectively add videos [Ctrl+insert], delete imported videos [Ctrl+del], change the order of the videos [ctrl+arrow up] and [ctrl+arrow down] and show the video properties. The button to the far right contains properties for the video (start/end time, frame rate, file name and resolution).

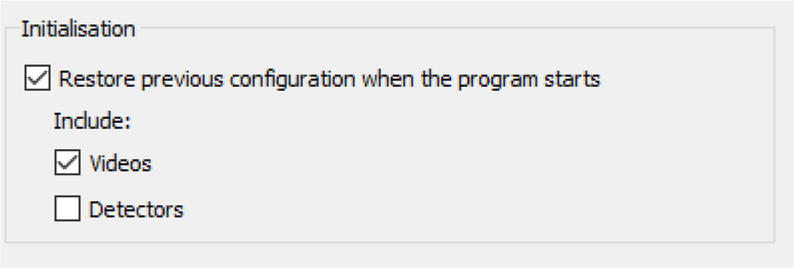
18. Inserted/created detectors.

19. Support tools for creating and calibrating of detectors. From left: 1) edit the detector [ctrl+R]. 2) delete the marked detector [Del]. 3) reset the detector. 4) histograms [F5]. 5) Overlay of the detector in the image [F6]. 6) information about activity in the detector [F7]. 7) extended information about the detector [F8].

20. Video pane.

4 SETTINGS

Access the settings under File -> Settings and a settings panel similar to Figure I.15 will be shown.



Initialisation

☑ Restore previous configuration when the program starts

Include:

☑ Videos

□ Detectors

Fig. I.16: The general settings in RUBA.

4.1 General

The general settings pane is shown in Figure I.16.

- **Restore previous configuration when the program starts:** When you open RUBA it will try to load the videos that you imported the last time you were running RUBA. You can change this behaviour or choose to also load the detectors from the last time you were running RUBA.
- An alternative is to use the `Load configuration` and `Save current configuration` buttons in the `File` menu. A configuration file is, similar to the detector files, an .yml-file, but contains references to the videos and detectors that are currently loaded into RUBA. By using this functionality, you can quickly switch between different combinations of videos and detectors.

4.2 Video synchronisation

The video synchronization pane is shown in Figure I.16.

- **Start time of each video is encoded in the file name:** If the frame rate of the video is constant, the start time of the video might be encoded into the file name of the video. The exact time of a frame will be computed based on the start time, the frame rate of the video, and the current frame number.
  - **Frame rate:** Used to define the frame rate. The value should match the frame rate of which the video is recorded. It is recommended to let RUBA auto-detect the frame rate (default).
  - **Date and time:** Used to define the date and time of which the video is recorded. This can be encoded in the file name, so that the information can be imported automatically. The format is chosen as either:

    - ∗ `MM-dd-HH` (month-day-hour). The year must be specified manually when playing the video.
    - ∗ `yyyy-MM-dd` (year-month-day)
    - ∗ `yyyyMMdd-HH` (year-month-day-hour)
    - ∗ `yyyy-MM-dd-HH` (year-month-day-hour)
    - ∗ `yyyyMMdd-HH-mm-ss` (year month day-hour-minute-second)
    - ∗ `yyyyMMdd-HH-mm-ss.zzz`
      (year month day-hour-minute-second.millisecond)
    - ∗ `user defined` in which the date and time is specified manually every time the video is played from the beginning.

- **Time stamps of each video frame are provided in a separate log file:** If the frame rate of the video is varying, you may put the exact date and time of each frame in a separate log file. The log file should be placed in the same directory as the corresponding video file and share the same file name except for the extension.

  - As default, RUBA looks for corresponding files with the '.log' extension. You may change this if necessary.
  - Each line of the log file should be contain the frame number and the frame time in the following format: `frameNbr yyyy MM dd hh:mm:ss.zzz`. An example is shown in Figure I.2.

The video processing pane is shown in Figure I.17b.

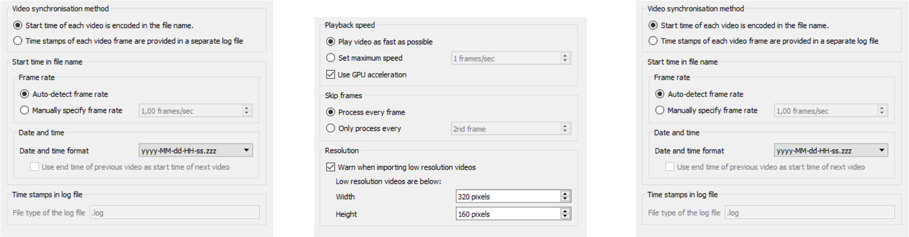
- **Playback speed:** Used to decide the speed of which the video will be analysed. The speed can be altered later on.

- **Skip frames:** In order to speed up processing, RUBA may skip every n'th frame. Beware, however, that this may affect the accuracy of the detectors.

- **Resolution:** Used to create a warning if the imported video is recorded at a low resolution. The width and height of when the warning is created can be set manually.

The behaviour of the built-in video recording may be altered from the settings pane shown in Figure I.17c. As default, RUBA records to a single file when the `Record video to file` button is pressed. However, when log files are played back with the Log File Reviewer, it might be beneficial to create a separate recording for each event.

If the option `Create a new recording for each jump of at least` is selected, a new recording will be created when the `jump to frame` functionality is used, either directly or indirectly from the Log File Reviewer. If this option is unchecked, a single video file will be created that contains the same 'jumps' as the original playback from RUBA.



Video synchronisation method

● Start time of each video is encoded in the file name.

○ Time stamps of each video frame are provided in a separate log file

Start time in file name

Frame rate

● Auto-detect frame rate

○ Manually specify frame rate 1,00 frames/sec

Date and time

Date and time format yyyy-MM-dd-HH-ss.zzz

□ Use end time of previous video as start time of next video

Time stamps in log file

File type of the log file .log

(a) The synchronisation settings in RUBA.

Playback speed

● Play video as fast as possible

○ Set maximum speed 1 frames/sec

☑ Use GPU acceleration

Skip frames

● Process every frame

○ Only process every 2nd frame

Resolution

☑ Warn when importing low resolution videos

Low resolution videos are below:

Width 320 pixels

Height 160 pixels

(b) The processing settings in RUBA.

Video synchronisation method

● Start time of each video is encoded in the file name.

○ Time stamps of each video frame are provided in a separate log file

Start time in file name

Frame rate

● Auto-detect frame rate

○ Manually specify frame rate 1,00 frames/sec

Date and time

Date and time format yyyy-MM-dd-HH-ss.zzz

□ Use end time of previous video as start time of next video

Time stamps in log file

File type of the log file .log

(c) The recording settings in RUBA.

Fig. I.17: Settings panels in RUBA.

5 DETECTOR TYPES

The algorithm behind the software consists of four detector types (presence, movement, stationary, and traffic light) with different attributes. Examples of the application of the four detector types are shown in Table I.3.

Combinations of the detectors are introduced on the description of the detector modules in Section 6.

An overview of the adjustable parameters for each detector type is given in Table I.4. A detailed description of the parameters is given with the overall description of each detector type.

5.1 Presence Detector

The *presence detector* checks if there is an object in a specific area of the video. With this method objects (vehicles, road users) that are not a part of the background (the road, the surroundings) are extracted. This is done by converting the image to a gray scale image and finding presences (continuous lines), where large variations in the contrast appear. In this way the algorithm finds road markings, changes in the pavement, and road users. This is done for all the frames of the video. For two consecutive frames, vectors between the lines are created and summed up. In this way we get a measure for the activity which is based partly on the size of the object, partly on the speed of the object moving across the area. In order to take noise in the image into consideration; i.e. from small changes in the contract, birds, movement of leaves, or shadows, the sensitivity of the presence detector is controlled by some parameters. The background is updated regularly to extract elements that are consistent in the image for a long time, or elements that occur due to changes in the light conditions and the creation of shadows.

Table I.2: A sample log file containing time stamps for every individual frame.

```
00000 2016 08 25 12:02:16.215
00001 2016 08 25 12:02:16.248
00002 2016 08 25 12:02:16.282
00003 2016 08 25 12:02:16.315
00004 2016 08 25 12:02:16.348
00005 2016 08 25 12:02:16.382
00006 2016 08 25 12:02:16.415
00007 2016 08 25 12:02:16.448
00008 2016 08 25 12:02:16.481
00009 2016 08 25 12:02:16.514
00010 2016 08 25 12:02:16.548
00011 2016 08 25 12:02:16.581
```

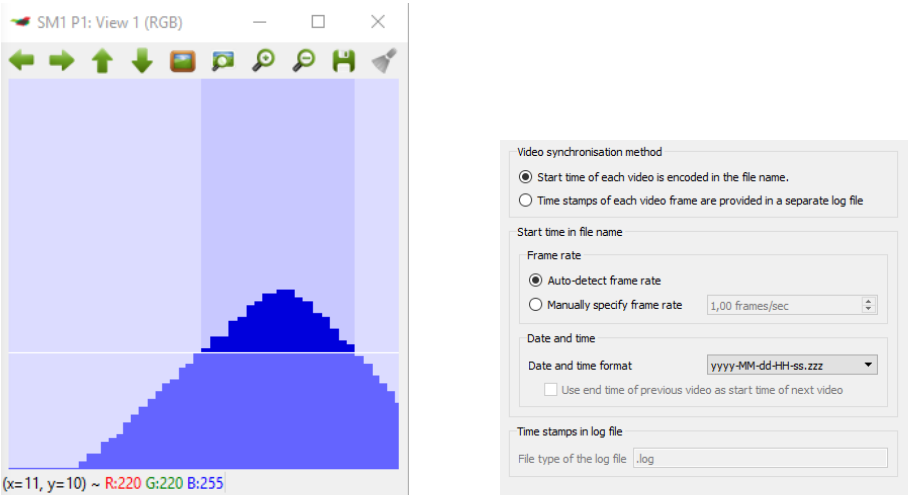
Table I.3: Applications of the detector types in RUBA.

Detector	Description
Presence	Simple analysis and traffic counts. Traffic counts in sections. NB! The highest accuracy is obtained if the traffic streams are separated.
Movement	Analysis and traffic counts in areas shared by road users from different directions (e.g. in intersections). Road users driving in the opposite direction of travel.
Stationary	Analysis of road users that do not move. Detection of parked cars.
Traffic light	Analysis of the phases of one or several traffic lights. Combined with other detectors, analysis of red light running.

Table I.4: Adjustable parameters in the detector modules.

	Presence	Movement	Stationary
Trigger threshold		x	
Minimum occupation percentage	x		x
Minimum speed		x	x
Movement direction		x	
Max vector count			x
Max triggered duration	x	x	x

Presence detector histogram



Video synchronisation method

● Start time of each video is encoded in the file name.

○ Time stamps of each video frame are provided in a separate log file

Start time in file name

Frame rate

● Auto-detect frame rate

○ Manually specify frame rate 1,00 frames/sec

Date and time

Date and time format yyyy-MM-dd-HH-ss.zzz

□ Use end time of previous video as start time of next video

Time stamps in log file

File type of the log file .log

(a) Histogram of the presence detector. X-axis: time. Y-axis: registered occupation percentage, in the range from 0 to 100.

(b) Movement detector histogram. X-axis: time. Y-axis: activity through the detector (the higher red lines, the more activity was registered

Fig. I.18: Histograms of the movement and presence detectors.

The *lower white, horizontal line* of the presence detector histogram of Figure I.18a shows if the size of the object that is present inside the detector is lower or higher than the **Minimum occupation percentage**.

To be detected as a road user, the blue lines must go higher than the horizontal line and the width of the blue lines above the threshold must be above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms). Only the dark blue lines counts in the time that it should be above the threshold to be detected as a road user. The width of the dark blue part should be approx. 0.5 cm when using the standard setting of 200 ms. If the activity for one road user results in two tops with a small gap in between (meaning that it will be registered twice), you can adjust the value for **Collate events within** (standard setting: 300 ms). In this way you can influence how large the gap between two tops can be before they should be registered as two separate road users. NB! Be careful if changing this value. If too big, two cars with a small gap between them will be registered as one.

Presence detector parameters

Minimum occupation percentage Fraction of the defined mask that must be occupied by a road user or a temporary object in the scene. Use this value to filter out noise or small road users, for instance pedestrians and bicyclists.

5.2 Movement Detector

The *movement detector* checks if there is activity in a specific direction in a certain area of the video by means of the *Farnebäck dense optical movement estimation*(Farnebäck, 2003). In the movement detector points in two consecutive frames are identified and matched. Objects moving across the detector will result in vectors that are dependent on the direction and speed of the object. Higher speeds will result in larger vectors. The amount of vectors per direction (in terms of degrees) is summed up for those vectors that are higher than a predefined value of the speed of the object. Vectors below this value are omitted.

Movement detector histogram

The histogram of the movement detector is illustrated in Figure I.18b. The *red, vertical lines* of the histogram indicate that something has moved faster than the **Minimum speed** in the specified direction. The unit is not comparable to known speed units (e.g. m/s or km/h) but low values means that slow road users will be detected, high values that only fast objects will be detected. Choosing a narrow range of direction will result in limited activity and hence few/short red lines.

The *white, horizontal line* shows the Trigger threshold, i.e. how much activity is required to be identified as a road user (above the line) and what is considered as noise (below the line). To be detected as a road user, the red lines must go higher than the horizontal line and the width of the red lines above the threshold must be above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms). Only the dark red

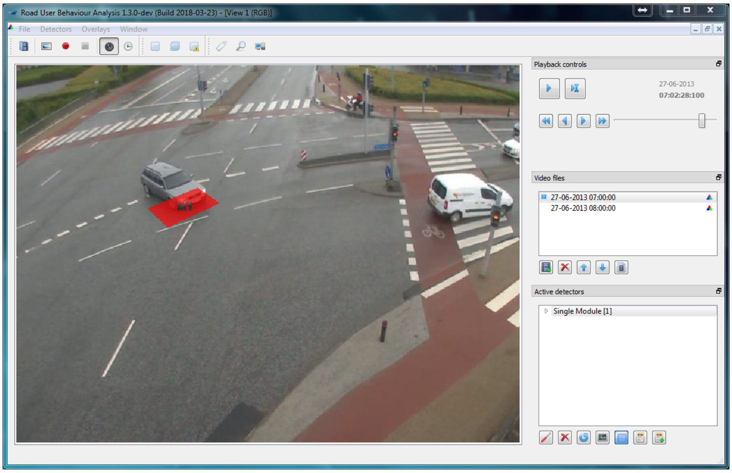


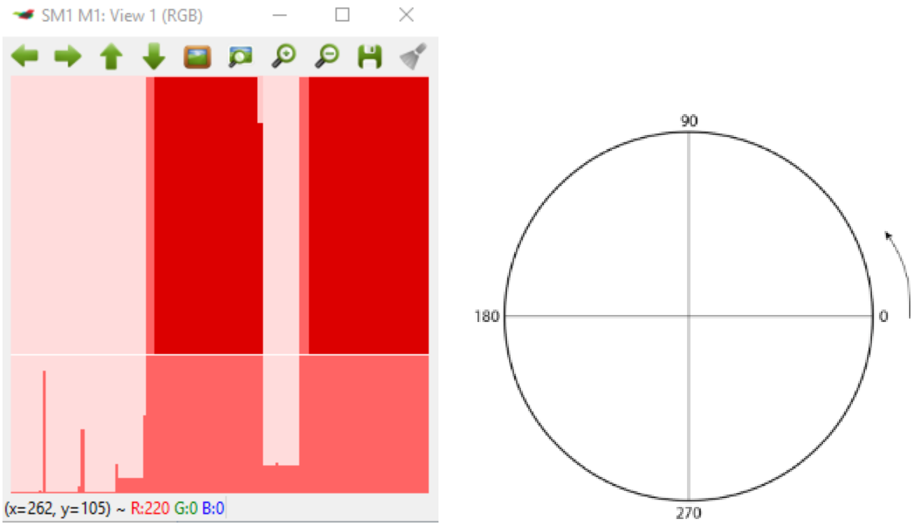
Fig. I.19: The movement detector (red). The movement detector detects activity (movement) in a specific direction through the detector.

lines counts in the time that it should be above the threshold to be detected as a road user. The width of the dark red part should be approx. 0.5 cm when using the standard setting of 200 ms. If the activity for one road user results in two tops with a small gap in between (meaning that it will be registered twice), you can adjust the value for **Collate events within** (standard setting: 300 ms). In this way you can influence how large the gap between two tops can be before they should be registered as two separate road users. NB! Be careful if changing this value. If too big, two cars with a small gap between them will be registered as one.

Movement detector parameters

Trigger threshold Limit for when an activity will be registered. The parameter is used to sort out noise in the video. In the histogram of Figure I.20a, the trigger threshold is shown as a horizontal line. To filter out noise, the red line must be above the horizontal line in four out of the last ten frames. This is visible from the histogram below; the red line is above the horizontal line in a short duration (three frames) before the detector is finally triggered and turns dark red.

Minimum speed Measure for how fast an object must move to be registered in the movement detector. The higher value, the faster it has to move. The minimum speed is measured in pixels.



(a) Trigger threshold

(b) Flow range

Fig. I.20: Histogram and flow range of the movement detector.

Flow range Defines in which direction the vectors must go if the activity should be registered as an event. The range can be chosen on a circle (0-360 degrees), illustrated in Figure I.20b. The range is chosen by either inserting the range in the fields or by dragging in the dots on the circles.

5.3 Stationary Detector

The *stationary detector*, depicted in Figure I.21, detects if an object is idling or moves very slowly through a specific area of the video by means of a combination of the presence and movement detectors. To detect an object (a road user) the presence detector must be triggered, while the movement detector must not be triggered. This indicates that an object is present in the area but is moving very slowly or not moving at all.

Stationary detector histogram

The histogram of the stationary detector, shown in Figure I.22, is a bit different than the histograms of the other detectors, as the detector is a combination of the presence (blue) and movement (red) detectors.

The *lower white, horizontal line* of the histogram shows if the size of the object that is present inside the detector is lower or higher than the **Minimum occupation percentage**.

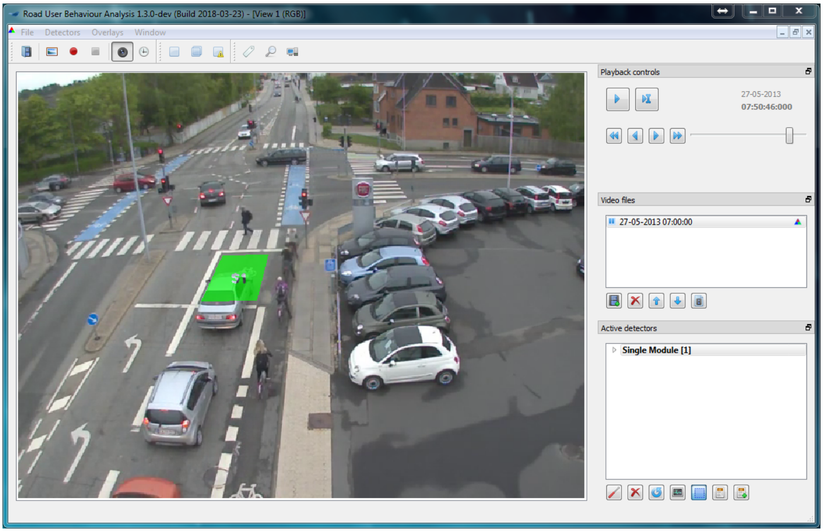


Fig. I.21: The stationary detector (green). The stationary detector detects if something is idling or moves slowly through the area covered by the detector

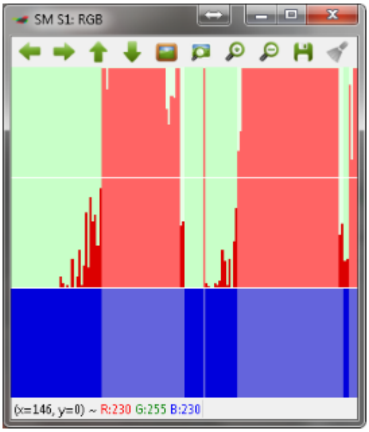


Fig. I.22: Stationary detector histogram. X-axis: time Y-axis: activity through the detector (the higher lines, the more activity was registered))

The *upper white, horizontal line* shows the limit of how much the object can move and still be registered as standing still. The height of the red line shows the amount of activity in any direction with a speed higher than the **Minimum speed**. To be detected as a road user standing still, the blue lines must reach the lower white, horizontal line and the red lines must not exceed the upper white, horizontal line. If both of these criteria are met, the blue and red lines will turn into a brighter colour. If the width of bright coloured lines is above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms), a road user is detected. The width of the dark red part should be approx. 0.5 cm when using the standard setting of 200 ms. If there is a small gaps in the bright coloured lines, it will still be registered as one road user if the gap is smaller than the value for **Collate events within** (standard setting: 300 ms). NB! Be careful if changing this value.

Stationary detector parameters

Minimum occupation percentage See the description of the equivalent parameter for the presence detector.

Minimum speed See the description of the equivalent parameter for the movement detector.

Max vector count The maximum amount of vectors that is allowed to be above the defined minimum speed to result in an event. In other words; the maximum allowed amount of 'movement' in the mask. If the movement is larger than this, we do not consider the mask to be stationary.

5.4 Traffic Light Detector

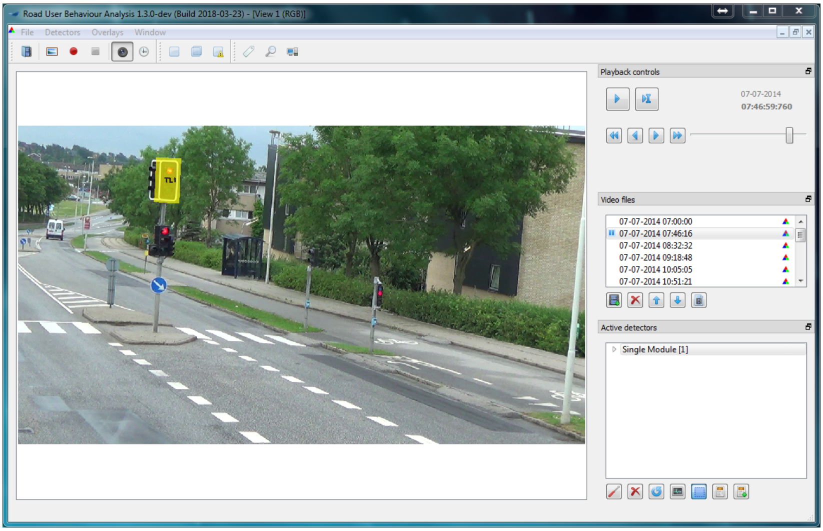
The *traffic light detector*, illustrated in Figure I.23, detects the different phases (red, yellow, red-yellow, and green) of a traffic light. The detector mask is to be defined in a small region around the traffic light and is used to perform image stabilisation. Image stabilisation is performed in order to make sure that the annotated traffic light positions follows the actual positions of the traffic light in case of small movements (oscillations) of the camera.

The traffic light positions should be annotated in the centre of the traffic light. An overview of the detected states of a traffic light is given in Table I.5.

Traffic light detector histogram

The histogram of the traffic light detector visualises the detected state of the traffic light. Two examples are shown in Figure I.24.

The five states of the traffic light are displayed in different colours that may be seen from the annotated histogram of Figure I.24b.



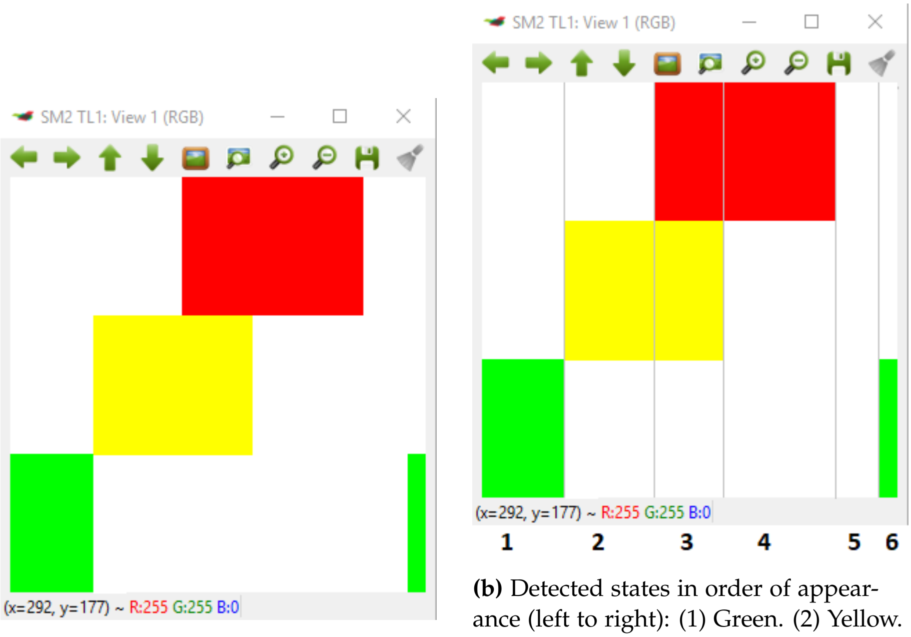
07-07-2014
07:46:59:760

07-07-2014 07:00:00
07-07-2014 07:46:16
07-07-2014 08:32:32
07-07-2014 09:18:48
07-07-2014 10:05:05
07-07-2014 10:51:21

Fig. I.23: The traffic light detector (yellow). The traffic light detector detects the colour of the traffic signal

Table I.5: Overview of the possible states of the traffic light and the corresponding detections as defined by the colour trigger. The ambiguous state is activated if the detector is unsure of the state of the traffic light.

Traffic light state	Colour trigger		
	Red	**Yellow**	**Green**
Red	x		
Red-yellow	x	x	
Yellow		x	
Green			x
Ambiguous			


SM2 TL1: View 1 (RGB)
(x=292, y=177) ~ R:255 G:255 B:0
SM2 TL1: View 1 (RGB)
(x=292, y=177) ~ R:255 G:255 B:0
1
2
3
4
5
6

(a) X-axis: time. Y-axis: detected state of the traffic light.

(b) Detected states in order of appearance (left to right): (1) Green. (2) Yellow. (3) Red-yellow. (4) Red. (5) Ambiguous (blank). (6) Green.

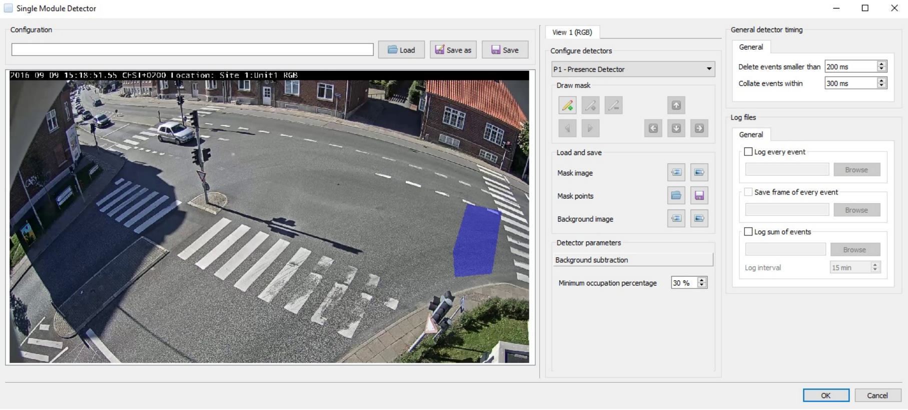
Fig. I.24: Histograms of the traffic light detector.

6 DETECTOR MODULES

The four detectors can be combined in detector modules. The modules manage logic between one or two detectors. Furthermore, the modules define when a detector takes on one of three states:

1. *Activated:* When a detector is activated, movement in the field can be registered.
2. *Triggered:* The detector has registered activity of the right type (e.g. the right direction) and in an extent that indicates that the movement comes from a road user (and not just noise in the image).
3. *Flagged* (results in the detection of an event) When the detector has been triggered for a number of consecutive frames, an event is registered and saved in a log file.

RUBA consists of a single module (one detector), a double module (two detectors), and an exclusive module (two detectors).



P1 - Presence Detector

Background subtraction

Minimum occupation percentage: 30 %

Delete events smaller than: 200 ms

Collate events within: 300 ms

☐ Log every event

☐ Save frame of every event

☐ Log sum of events

Log interval: 15 min

Fig. I.25: Example of a single module. The single module lets the user create one detector of one's own choice per module.

6.1 Single

The single module consists of one detector type (presence/movement/stationary-/traffic light). An event is saved in a log file when the criteria are met according to the specifications of the chosen detector. An example is shown in Figure I.25.

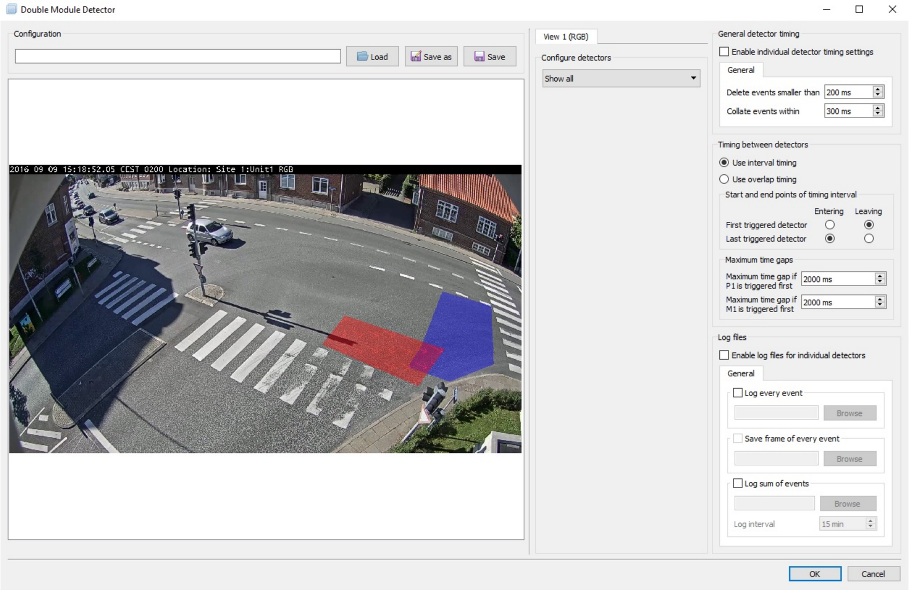
6.2 Double

Consists of two optional detectors. An event is detected and saved in the log file when both detectors have been triggered within a specific time distance defined by the user. An example is shown in Figure I.26.

The timewise relation between the two individual detectors can be defined in two ways. *Interval timing* can be used to define the maximum time gap between the two detectors are triggered or stop being triggered. For instance, the time can be defined as the the interval from a vehicle enters one detector and another vehicle enters the other detector. *Overlap timing* is used when the two detectors should be activated simultaneously. It is possible to define a buffer so that events can be registered if the detectors are activated almost at the same time. Detailed information on the timing options is given in Section 7.3.

6.3 Exclusive

Similar to the double module, the exclusive module consists of two optional detectors. An event is detected and saved in the log file only when the main detector is triggered and the excluding detector is not. If both detectors are



General detector timing

□ Enable individual detector timing settings

General

Delete events smaller than 200 ms

Collate events within 300 ms

Timing between detectors

● Use interval timing

○ Use overlap timing

Start and end points of timing interval

	Entering	Leaving
First triggered detector	○	●
Last triggered detector	●	○

Maximum time gaps

Maximum time gap if P1 is triggered first 2000 ms

Maximum time gap if M1 is triggered first 2000 ms

Log files

□ Enable log files for individual detectors

General

□ Log every event

□ Save frame of every event

□ Log sum of events

Log interval 15 min

Fig. I.26: Example of a double module. The double module lets the user create two detectors of one's own choice per module.

triggered, an event is not created. A timing example is shown in Figure I.27.

7 SETTING UP THE LOGGER

To set up the logger, two aspects should be considered; timing settings and the type of output we get from RUBA. The common timing settings are related to each individual detector. Furthermore, there are additional timing settings for double modules to define when an event should be detected.

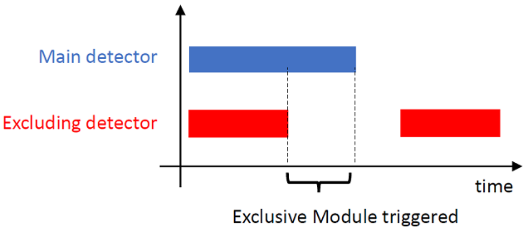
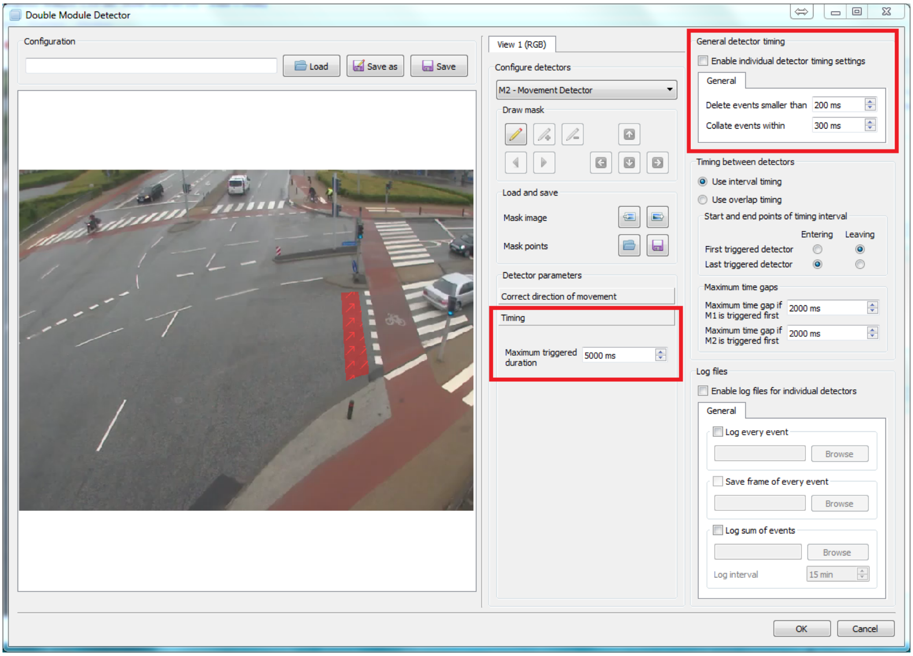


Fig. I.27: The Exclusive module is triggered when the main detector is triggered and the excluding detector is not.



M2 - Movement Detector

Maximum triggered duration: 5000 ms

□ Enable individual detector timing settings

Delete events smaller than: 200 ms

Collate events within: 300 ms

● Use interval timing

○ Use overlap timing

	Entering	Leaving
First triggered detector	○	●
Last triggered detector	●	○

Maximum time gap if M1 is triggered first: 2000 ms

Maximum time gap if M2 is triggered first: 2000 ms

□ Enable log files for individual detectors

□ Log every event

□ Save frame of every event

□ Log sum of events

Log interval: 15 min

Fig. I.28: Common timing settings.

7.1 Common timing settings

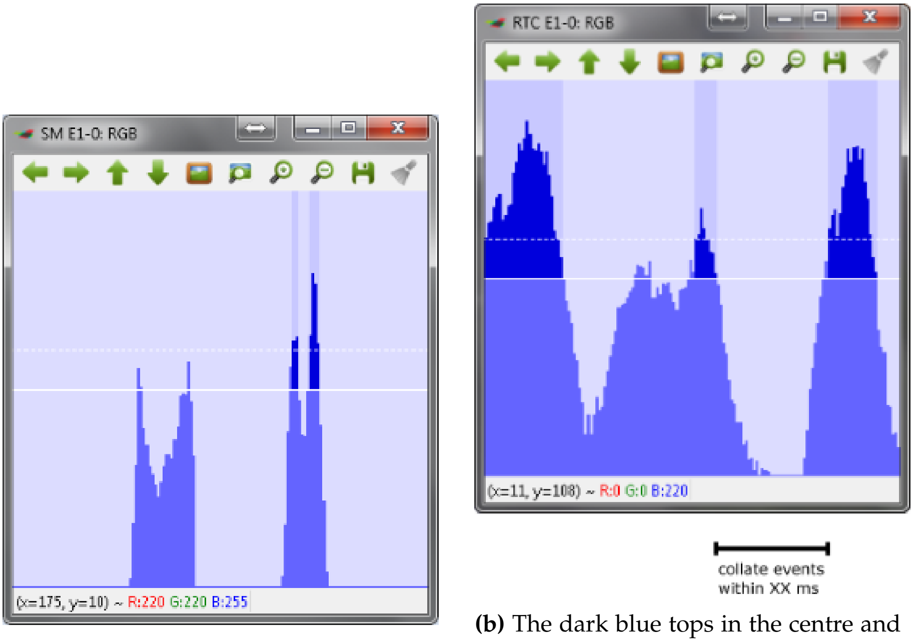
The common timing settings are used to adjust when an event should be written to the log. The fields for defining the common timing settings are marked in red on Figure I.28.

Delete events smaller than

Deletes events that are only detected briefly, which often indicates noise in the image. Events with duration less than `Delete events smaller than` milliseconds will be omitted from the log. An example is shown in Figure I.29a.

Collate events within

Combines separate events into one event if the time gap between them is less than XX milliseconds. This protects against multiple detections of the same object. If chosen too high, multiple road users driving close to each other will be registered as one road user. See Figure I.29b for an example.


SM E1-0: RGB
(x=175, y=10) ~ R:220 G:220 B:255
RTC E1-0: RGB
(x=11, y=108) ~ R:0 G:0 B:220
collate events
within XX ms

(a) The two dark blue tops in the centre of the histogram will not be detected as an event if `Delete events smaller than` is greater than zero.

(b) The dark blue tops in the centre and the right of the image might be grouped as one event if the `Collate events within` setting is greater than the time difference between the two detections.

Fig. I.29: The common timing settings in RUBA.

Maximum triggered duration

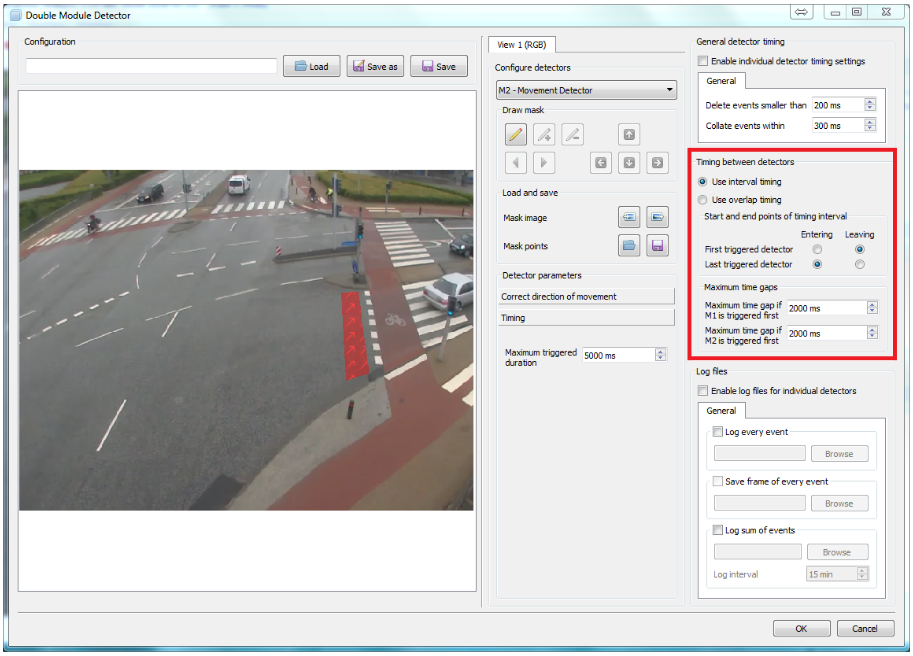
Defines the maximum allowed duration (in milliseconds) of an event. If an event is longer than the specified maximum duration, it will be cut off after the max triggered duration has gone, and a new event will be created immediately thereafter.

7.2 Double module timing settings

Timing between the two individual detectors of a double module can be defined using either *interval timing* or *overlap timing*, marked in red on Figure I.30.

Interval timing

The interval timing, illustrated in Figure I.31a, denotes the maximum accepted time gap (in milliseconds) from the detection of activity in one detector to the detection of activity in the other detector. The point for measuring



Configuration

Configure detectors: M2 - Movement Detector

Detector parameters

Maximum triggered duration: 5000 ms

General detector timing

☐ Enable individual detector timing settings

Delete events smaller than: 200 ms

Collate events within: 300 ms

Timing between detectors

● Use interval timing

○ Use overlap timing

Start and end points of timing interval

	Entering	Leaving
First triggered detector	○	●
Last triggered detector	●	○

Maximum time gaps

Maximum time gap if M1 is triggered first: 2000 ms

Maximum time gap if M2 is triggered first: 2000 ms

Log files

☐ Enable log files for individual detectors

☐ Log every event

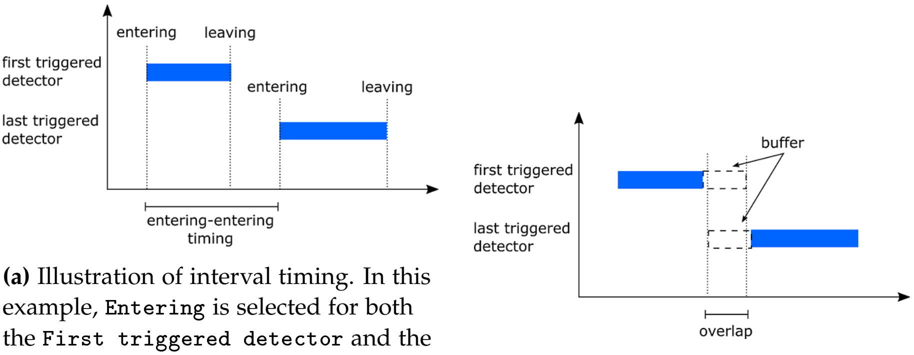
☐ Save frame of every event

☐ Log sum of events

Log interval: 15 min

Fig. I.30: The Double Module offers two timing modes; interval timing and overlap timing.

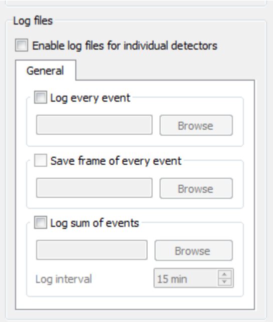
the time gap can be either when the detector is activated (i.e. when the detector registers a that a road user enters the detector) or when the detector is left (i.e. when the road user has just left the detector).



(a) Illustration of interval timing. In this example, `Entering` is selected for both the `First triggered detector` and the `Last triggered detector`.

(b) Illustration of overlap timing.

Fig. I.31: Timing settings of the Double Module.



Log files

☐ Enable log files for individual detectors

☐ Log every event

☐ Save frame of every event

☐ Log sum of events

Log interval 15 min

Fig. I.32: The log files settings pane.

Overlap timing

The overlap timing, illustrated in Figure I.31b, detects an event if both detectors are activated simultaneously. A buffer (in milliseconds) can be used to also log events where the two detectors are activated at almost the same time.

7.3 Logs

Three types of output can be created:

- **Log every event**: creates a .csv file with one line for each detection.
- **Save frame of every event**: saves an image of what triggered the detector. For double modules, the saved image contains a screenshot of what triggered each detector, put side by side.
- **Log sum of events**: creates a .csv file with the total number of detections per log interval(in minutes).

The log settings pane is shown in Figure I.32. The content of the log file depends on the detector module. Table I.6 gives an overview of the content of the log files.

Log Examples

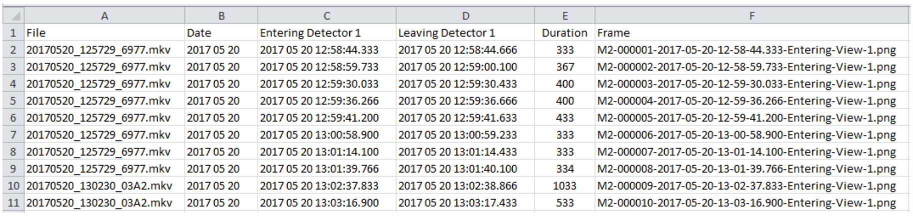
Sample every event logs are shown in Figure I.33 and I.34. A sample of a sum of event log is shown in Figure I.35.

8 GROUND TRUTH ANNOTATOR

RUBA features a built-in option to perform manual event-based annotation based on timestamps. For example, we might want to annotate whenever a

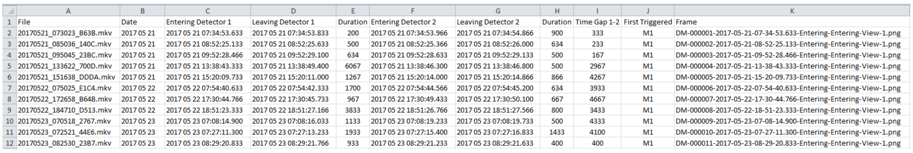
Table I.6: Contents in the log files. The files `Analytics` are created when marking `Log every event` in the settings/drawing window of the detector, while `Counts` will be created when marking `Log sum of events`. In the latter, the desired time interval can be specified (in minutes), e.g. 15 minutes. *Legend*: Single Module = SM, Double Module = DM, Exclusive Module = XM.

	Log every event		Log sum of events	Description
	SM, XM	DM	All modules	
File	x	x	x	File name of the video
Date	x	x	x	Date for video recording
Entering	x	x		Time stamp for arrival to the detector (i.e. there is activity in the detector)
Leaving	x	x		Time stamp for when the detector has been left (i.e. is empty again)
Timegap 1-2		x		Time difference between an object has triggered detector 1 and an object has triggered detector 2
Timegap DetectorX		x		Time from one object arrives to detector X (trigger the detector) to the road user has left the detector
FirstTriggered		x		Which of the two detectors was triggered first?
TimeStart			x	Time for the beginning of the time interval
TimeEnd			x	Time for the end of the time interval
Object			x	Number of objects that have been detected within a specific time interval



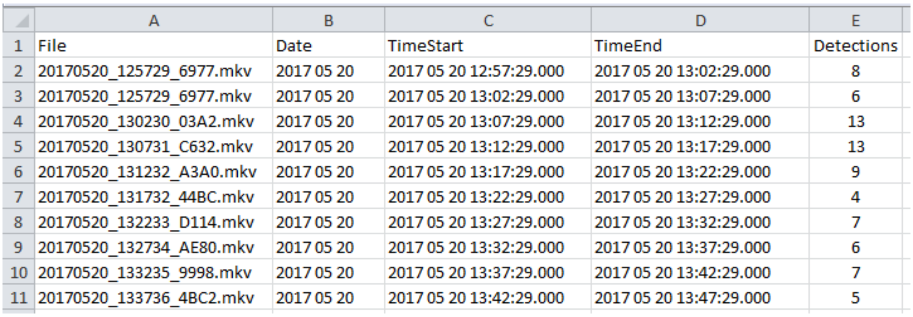
	A	B	C	D	E	F
1	File	Date	Entering Detector 1	Leaving Detector 1	Duration	Frame
2	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:58:44.333	2017 05 20 12:58:44.666	333	M2-000001-2017-05-20-12-58-44.333-Entering-View-1.png
3	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:58:59.733	2017 05 20 12:59:00.100	367	M2-000002-2017-05-20-12-58-59.733-Entering-View-1.png
4	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:30.033	2017 05 20 12:59:30.433	400	M2-000003-2017-05-20-12-59-30.033-Entering-View-1.png
5	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:36.266	2017 05 20 12:59:36.666	400	M2-000004-2017-05-20-12-59-36.266-Entering-View-1.png
6	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:41.200	2017 05 20 12:59:41.633	433	M2-000005-2017-05-20-12-59-41.200-Entering-View-1.png
7	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:00:58.900	2017 05 20 13:00:59.233	333	M2-000006-2017-05-20-13-00-58.900-Entering-View-1.png
8	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:01:14.100	2017 05 20 13:01:14.433	333	M2-000007-2017-05-20-13-01-14.100-Entering-View-1.png
9	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:01:39.766	2017 05 20 13:01:40.100	334	M2-000008-2017-05-20-13-01-39.766-Entering-View-1.png
10	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:02:37.833	2017 05 20 13:02:38.866	1033	M2-000009-2017-05-20-13-02-37.833-Entering-View-1.png
11	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:03:16.900	2017 05 20 13:03:17.433	533	M2-000010-2017-05-20-13-03-16.900-Entering-View-1.png

Fig. I.33: Every event log from a Single Module Detector. The `Duration` column lists the time difference in milliseconds between the `Entering Detector 1` and `Leaving Detector 1`.



	A	B	C	D	E	F	G	H	I	J	K
1	File	Date	Entering Detector 1	Leaving Detector 1	Duration	Entering Detector 2	Leaving Detector 2	Duration	Time Gap 1-2	First Triggered	Frame
2	20170521_073023_B63B.mkv	2017 05 21	2017 05 21 07:34:53.633	2017 05 21 07:34:53.833	200	2017 05 21 07:34:53.966	2017 05 21 07:34:54.866	900	333	M1	DM-000001-2017-05-21-07-34-53.633-Entering-Entering-View-1.png
3	20170521_085036_140C.mkv	2017 05 21	2017 05 21 08:52:25.133	2017 05 21 08:52:25.633	500	2017 05 21 08:52:25.366	2017 05 21 08:52:26.000	634	233	M1	DM-000002-2017-05-21-08-52-25.133-Entering-Entering-View-1.png
4	20170521_095045_23BC.mkv	2017 05 21	2017 05 21 09:52:28.466	2017 05 21 09:52:29.100	634	2017 05 21 09:52:28.633	2017 05 21 09:52:29.133	500	167	M1	DM-000003-2017-05-21-09-52-28.466-Entering-Entering-View-1.png
5	20170521_133622_700D.mkv	2017 05 21	2017 05 21 13:38:43.333	2017 05 21 13:38:49.400	6067	2017 05 21 13:38:46.300	2017 05 21 13:38:46.800	500	2967	M1	DM-000004-2017-05-21-13-38-43.333-Entering-Entering-View-1.png
6	20170521_151638_DDDA.mkv	2017 05 21	2017 05 21 15:20:09.733	2017 05 21 15:20:11.000	1267	2017 05 21 15:20:14.000	2017 05 21 15:20:14.866	866	4267	M1	DM-000005-2017-05-21-15-20-09.733-Entering-Entering-View-1.png
7	20170522_075025_E1C4.mkv	2017 05 22	2017 05 22 07:54:40.633	2017 05 22 07:54:42.333	1700	2017 05 22 07:54:44.566	2017 05 22 07:54:45.200	634	3933	M1	DM-000006-2017-05-22-07-54-40.633-Entering-Entering-View-1.png
8	20170522_172658_B64B.mkv	2017 05 22	2017 05 22 17:30:44.766	2017 05 22 17:30:45.733	967	2017 05 22 17:30:49.433	2017 05 22 17:30:50.100	667	4667	M1	DM-000007-2017-05-22-17-30-44.766-Entering-Entering-View-1.png
9	20170522_184710_D513.mkv	2017 05 22	2017 05 22 18:51:23.333	2017 05 22 18:51:27.166	3833	2017 05 22 18:51:26.766	2017 05 22 18:51:27.566	800	3433	M1	DM-000008-2017-05-22-18-51-23.333-Entering-Entering-View-1.png
10	20170523_070518_2767.mkv	2017 05 23	2017 05 23 07:08:14.900	2017 05 23 07:08:16.033	1133	2017 05 23 07:08:19.233	2017 05 23 07:08:19.733	500	4333	M1	DM-000009-2017-05-23-07-08-14.900-Entering-Entering-View-1.png
11	20170523_072521_44E6.mkv	2017 05 23	2017 05 23 07:27:11.300	2017 05 23 07:27:13.233	1933	2017 05 23 07:27:15.400	2017 05 23 07:27:16.833	1433	4100	M1	DM-000010-2017-05-23-07-27-11.300-Entering-Entering-View-1.png
12	20170523_082530_23B7.mkv	2017 05 23	2017 05 23 08:29:20.833	2017 05 23 08:29:21.766	933	2017 05 23 08:29:21.233	2017 05 23 08:29:21.633	400	400	M1	DM-000011-2017-05-23-08-29-20.833-Entering-Entering-View-1.png

Fig. I.34: Every event log from a Double Module Detector. In the `First Triggered` column, we see that the Movement Detector 1 (M1) always is triggered first in this sample. The `Frame` column shows the file name of the corresponding snapshot image for each event. If this column is empty, the `Save frame of every event` checkbox has not been ticket in the log settings.

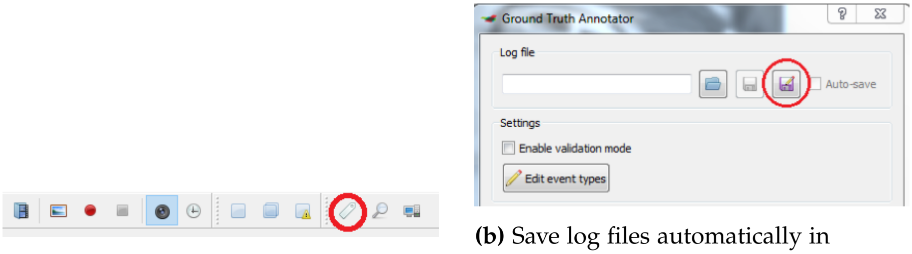


	A	B	C	D	E
1	File	Date	TimeStart	TimeEnd	Detections
2	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:57:29.000	2017 05 20 13:02:29.000	8
3	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:02:29.000	2017 05 20 13:07:29.000	6
4	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:07:29.000	2017 05 20 13:12:29.000	13
5	20170520_130731_C632.mkv	2017 05 20	2017 05 20 13:12:29.000	2017 05 20 13:17:29.000	13
6	20170520_131232_A3A0.mkv	2017 05 20	2017 05 20 13:17:29.000	2017 05 20 13:22:29.000	9
7	20170520_131732_44BC.mkv	2017 05 20	2017 05 20 13:22:29.000	2017 05 20 13:27:29.000	4
8	20170520_132233_D114.mkv	2017 05 20	2017 05 20 13:27:29.000	2017 05 20 13:32:29.000	7
9	20170520_132734_AE80.mkv	2017 05 20	2017 05 20 13:32:29.000	2017 05 20 13:37:29.000	6
10	20170520_133235_9998.mkv	2017 05 20	2017 05 20 13:37:29.000	2017 05 20 13:42:29.000	7
11	20170520_133736_4BC2.mkv	2017 05 20	2017 05 20 13:42:29.000	2017 05 20 13:47:29.000	5

Fig. I.35: Sum of event log. All detector modules produce sum logs in this format.

car turns right in an intersection or whenever a pedestrian passes a specific line in a zebra crossing. In the example below, we will set up the Ground Truth Annotator to perform annotation of different road users at an intersection.

Click on Ground Truth Annotator in the main RUBA menu, marked with red on Figure I.37a.



□ Auto-save

□ Enable validation mode

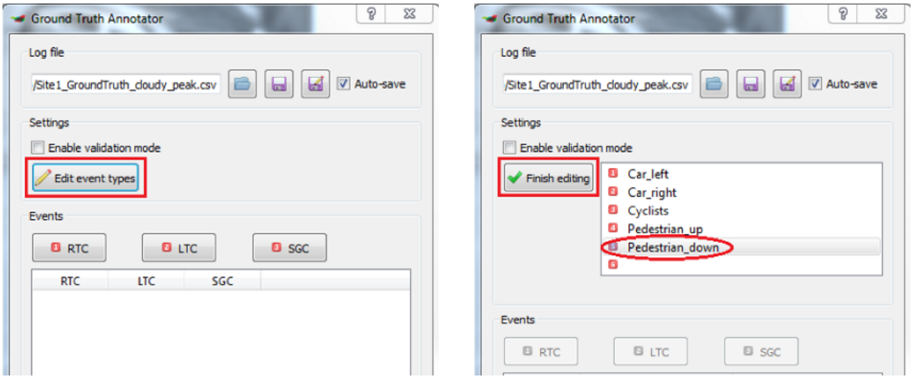
(a) Opening the Ground Truth Annotator.

(b) Save log files automatically in Ground Truth Annotator by pressing the `Auto-save` button.

Fig. I.36: Ground Truth Annotator.

A new window opens. Click on `Save log file as`, shown in Figure I.37b, and specify the name of the file and where to save the log file. Put a check mark next to Auto-save to save the log automatically.

Click on Edit event types to specify the types of road users to register. Double click on the name to change, and press `Finish editing` to include the event types in the `Event` panel. This process is also illustrated in Figure I.37.



Log file: /Site1_GroundTruth_cloudy_peak.csv ☑ Auto-save

□ Enable validation mode

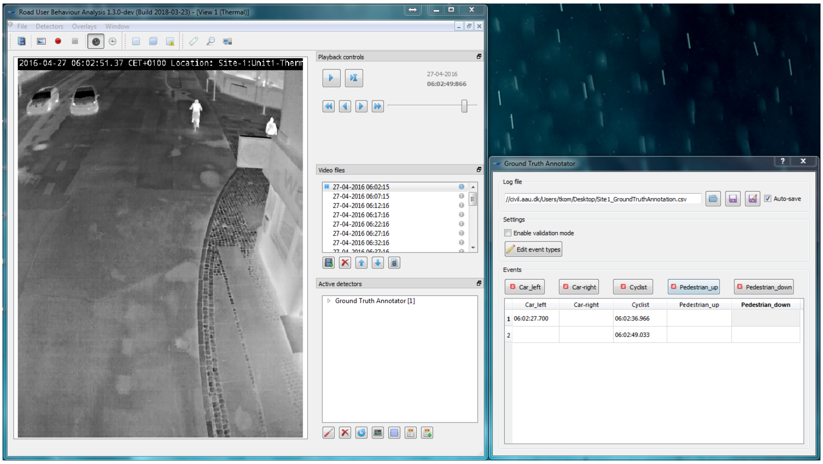
RTC	LTC	SGC

Car_left
Car_right
Cyclists
Pedestrian_up
Pedestrian_down

(a) **(b)**

Fig. I.37: Editing the event types of the Ground Truth Annotator

Adjust the window and column sizes to get it to look more clear and place



2016-04-27 06:02:51.37 CET+0100 Location: Site-1:Unit1-Therm

27-04-2016
06:02:49:866

27-04-2016 06:02:15
27-04-2016 06:07:15
27-04-2016 06:12:16
27-04-2016 06:17:16
27-04-2016 06:22:16
27-04-2016 06:27:16
27-04-2016 06:32:16

Ground Truth Annotator [1]

//civil.aau.dk/Users/tkom/Desktop/Site1_GroundTruthAnnotation.csv ☑ Auto-save

□ Enable validation mode

Edit event types

	Car_left	Car-right	Cyclist	Pedestrian_up	Pedestrian_down
1	06:02:27.700		06:02:36.966		
2			06:02:49.033		

Fig. I.38: The Ground Truth Annotator should be placed beside the video window in RUBA to make the annotation process feasible.

the window as shown in Figure I.38 so that you can see the window and the main window of RUBA at the same time.

Click in the Ground Truth Annotator (GTA) window and press space to start. Press space again to pause playback. Click on the corresponding button in the GTA window or press the number on the keyboard every time a road user of that particular type passes. Make sure that you register all road users at the same spot every time. All road users must be counted individually in the tool, so if there is a group of 2 pedestrians, press twice to register the road users.

8.1 Reviewing annotations

You may continue the annotation process or review the previous annotations by loading them into the Ground Truth Annotator. Either start the video from the beginning or double-click on an annotated time stamp to jump to this particular point in the video.

If you tick the `Enable validation mode` button during playback, the most recent annotated time stamp will be selected in the `Events` pane. If the `Show masks` button is also enabled in the main RUBA window, the annotated event information will be overlayed on the video when it takes place.

9 LOG FILE REVIEWER

The Log File Reviewer is a tool to review and validate the log files as output by the detector modules of RUBA. Open the Log File Reviewer by pressing

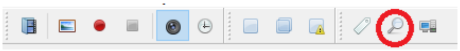
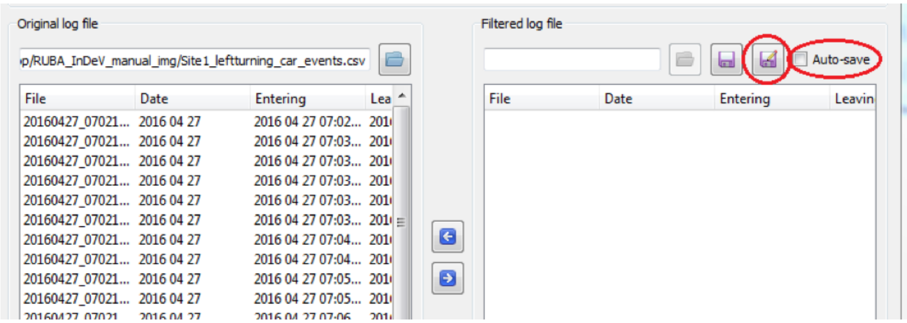


Fig. I.39: Opening the Log File Reviewer.



p/RUBA_InDeV_manual_img/Site1_leftturning_car_events.csv

File	Date	Entering	Lea
20160427_07021...	2016 04 27	2016 04 27 07:02...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021	2016 04 27	2016 04 27 07:06	201

File	Date	Entering	Leavin

Fig. I.40: Setting the file path of the filtered log file.

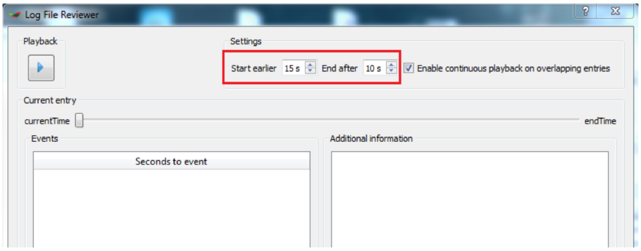
the button on the toolbar, illustrated in Figure I.39, or press F11.

Open the log file that you want to inspect by clicking the folder icon in the `Original log file` window, shown on the left of Figure I.40. Hereafter, click on the save button in the `Filtered log file window` and specify where to save the (validated) log file for further processing. Tick off the check box `Auto-save`.

Define how many seconds should be played before and after the time stamp, e.g. 15 sec before the `Entering` and 10 sec after the `Leaving` timestamps in the log files. An example is shown in Figure I.41.

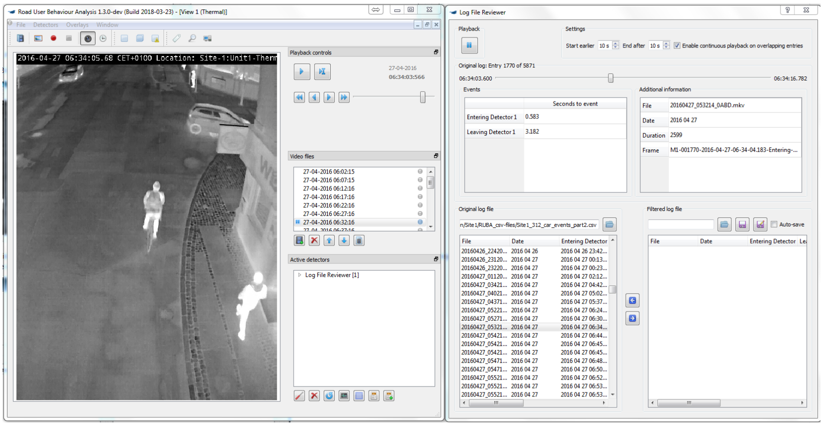
Adjust the size of the RUBA windows as illustrated in Figure I.42 so that the normal window and the log file reviewer are both visible and placed next to each other.

Double click on the first video in the list, then click the playback button, marked in red on Figure I.43a. All events will be played one by one without break. Press the pause button to pause the playback (keyboard shortcut:



Start earlier 15 s End after 10 s ☑ Enable continuous playback on overlapping entries

Fig. I.41: Setting the video playback properties.



Video files:

27-04-2016 06:02:15
27-04-2016 06:07:15
27-04-2016 06:12:16
27-04-2016 06:17:16
27-04-2016 06:22:16
27-04-2016 06:27:16
27-04-2016 06:32:16

Start earlier 10 s End after 10 s ☑ Enable continuous playback on overlapping entries

Original log: Entry 1770 of 5871

06:34:03.600 06:34:16.782

	Seconds to event
Entering Detector 1	0.583
Leaving Detector 1	3.182

File	20160427_053214_0ABD.mkv
Date	2016 04 27
Duration	2599
Frame	M1-001770-2016-04-27-06-34-04.183-Entering-...

n/Site1/RUBA_csv-files/Site1_312_car_events_part2.csv

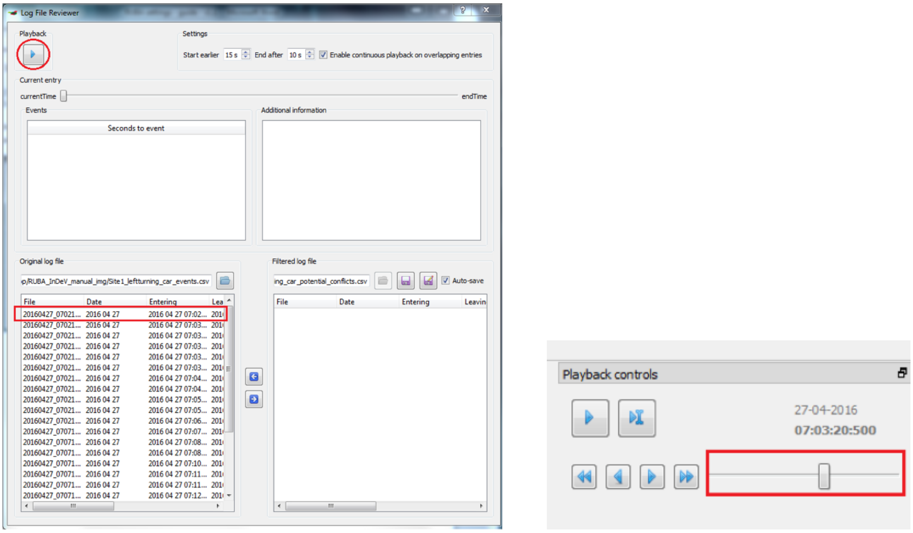
File	Date	Entering Detector
20160426_22420...	2016 04 26	2016 04 26 23:42...
20160426_23120...	2016 04 27	2016 04 27 00:13...
20160426_23220...	2016 04 27	2016 04 27 00:23...
20160427_01120...	2016 04 27	2016 04 27 02:12...
20160427_03421...	2016 04 27	2016 04 27 04:42...
20160427_04021...	2016 04 27	2016 04 27 05:02...
20160427_04371...	2016 04 27	2016 04 27 05:37...
20160427_05221...	2016 04 27	2016 04 27 06:24...
20160427_05271...	2016 04 27	2016 04 27 06:30...
20160427_05321...	2016 04 27	2016 04 27 06:34...
20160427_05421...	2016 04 27	2016 04 27 06:44...
20160427_05421...	2016 04 27	2016 04 27 06:45...
20160427_05421...	2016 04 27	2016 04 27 06:45...
20160427_05471...	2016 04 27	2016 04 27 06:48...
20160427_05471...	2016 04 27	2016 04 27 06:50...
20160427_05521...	2016 04 27	2016 04 27 06:52...
20160427_05521...	2016 04 27	2016 04 27 06:53...
20160427_05521...	2016 04 27	2016 04 27 06:53...

□ Auto-save

File	Date	Entering Detector	Le

Fig. I.42: The Log File Reviewer next to the main RUBA window.

space) and press again to start the playback (keyboard shortcut: space). You can double click on any of the events in the list to go to that event.



Start earlier 15 s End after 10 s ☑ Enable continuous playback on overlapping entries

Seconds to event

p/RUBA_InDeV_manual_img/Site1_leftturning_car_events.csv

File	Date	Entering	Lea
20160427_07021...	2016 04 27	2016 04 27 07:02...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021...	2016 04 27	2016 04 27 07:06...	201
20160427_07071...	2016 04 27	2016 04 27 07:07...	201
20160427_07071...	2016 04 27	2016 04 27 07:08...	201
20160427_07071...	2016 04 27	2016 04 27 07:08...	201
20160427_07071...	2016 04 27	2016 04 27 07:10...	201
20160427_07071...	2016 04 27	2016 04 27 07:11...	201
20160427_07071...	2016 04 27	2016 04 27 07:11...	201
20160427_07071...	2016 04 27	2016 04 27 07:12...	201

ing_car_potential_conflicts.csv ☑ Auto-save

File	Date	Entering	Leavin

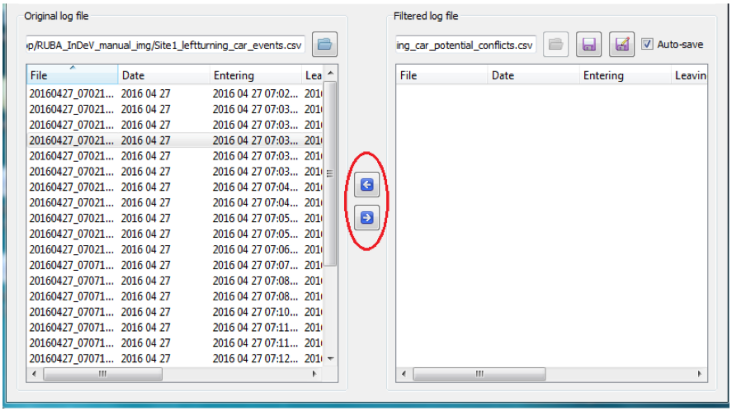
(a) Starting the playback based on the events of the log file.

(b) Adjusting the playback speed in the main RUBA window.

Fig. I.43: Playback in the Log File Reviewer.

Adjust the speed on the sliding bar (Figure I.43b) if it goes too fast/slow. Do the following to insert and delete events in the filtered log file:

1. Click on the blue arrow pointing right (keyboard shortcut: INS) to put



Original log file

p/RUBA_InDeV_manual_img/Site1_leftturning_car_events.csv

File	Date	Entering	Lea
20160427_07021...	2016 04 27	2016 04 27 07:02...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:03...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:04...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021...	2016 04 27	2016 04 27 07:05...	201
20160427_07021...	2016 04 27	2016 04 27 07:06...	201
20160427_07071...	2016 04 27	2016 04 27 07:07...	201
20160427_07071...	2016 04 27	2016 04 27 07:08...	201
20160427_07071...	2016 04 27	2016 04 27 07:08...	201
20160427_07071...	2016 04 27	2016 04 27 07:10...	201
20160427_07071...	2016 04 27	2016 04 27 07:11...	201
20160427_07071...	2016 04 27	2016 04 27 07:11...	201
20160427_07071...	2016 04 27	2016 04 27 07:12...	201

Filtered log file

ing_car_potential_conflicts.csv

☑ Auto-save

File	Date	Entering	Leavin

Fig. I.44: Inserting and deleting events into the filtered log file.

an event from the original when there is a potential conflict. It is possible to select more events at once.

2. Click on the blue arrow pointing left (keyboard shortcut: DEL) to remove an event from the filtered log file. These arrows are marked in red on Figure I.44.

Please note that only the filtered log file is altered during this process. The original log file is read-only.

Paper J

A Framework for Automated Traffic Analysis of Surrogate Measures of Safety From Video Using Deep Learning Techniques

Morten B. Jensen, Martin Ahrnbom, Maarten Kruithof, Kalle Åström, Mikael Nilsson, Håkan Ardö, Aliaksei Laureshyn, Carl Johnsson, and Thomas B. Moeslund

The paper has been accepted in the
Transportation Research Board (TRB) 98th Annual Meeting, In Press, 2019.

The layout has been revised.

ABSTRACT

Traffic surveillance and monitoring are gaining a lot of attention as a result of an increase of vehicles on the road and a desire to minimize accidents. In order to minimize accidents and near-accidents, it is important to be able to judge the safety of a traffic environment. It is possible to perform traffic analysis using large quantities of video data. Computer vision is a great tool for reducing the data, so that only sequences of interest are further analyzed. In this paper, we propose a cross-disciplinary framework for performing automated traffic analysis, from both a computer vision researcher's and traffic researcher's point-of-view. Furthermore, we present STRUDL, an open-source implementation of this framework, that computes trajectories of road users, which we use to automatically find sequences containing critical events of vehicles and vulnerable road users in an traffic intersection, which is an otherwise time-consuming task.

Keywords: Computer vision, data reduction, computer aided analysis, deep learning, surveillance, tracking, detection, traffic analysis

1 INTRODUCTION

In 2017 more than 25,000 people died and approximately 135,000 people were seriously injured on the roads in the European Union (EU) [1]. While the numbers are still very high, both injuries and fatalities have been decreasing for decades. Paradoxically, road safety experts worry about the problem of "too few crashes", referring to the difficulties using the traditional safety diagnosing methods as crash counts registered at individual sites become very low [2]. The situation is aggravated by the unresolved problems of crash under-reporting, scarce information about the crash details, conditions in standard police reports and the general retro-active nature of the crash analysis (before safety problem can be diagnosed, it has to manifest itself in form of crashes with people killed or injured).

A complementary approach to crash analysis is to use surrogate measures of safety (SMoS). The method rests on the assumption of a continuous relation between the severity of events in traffic and their frequency [3], visualized in Figure J.1. The fatal and injury crashes are the most severe events and occur relatively seldom, while the events of "normal" severity can be observed in hundreds or thousands every day. The SMoS are normally derived from non-crash events that are close enough to crashes on the severity scale to possess sufficient similarities and thus be relevant for the safety, but much more frequent compared the actual crashes.

While the idea has been known for decades [4, 5], the lack of an efficient tool to reliably and accurately measure SMoS hindered the method from be-

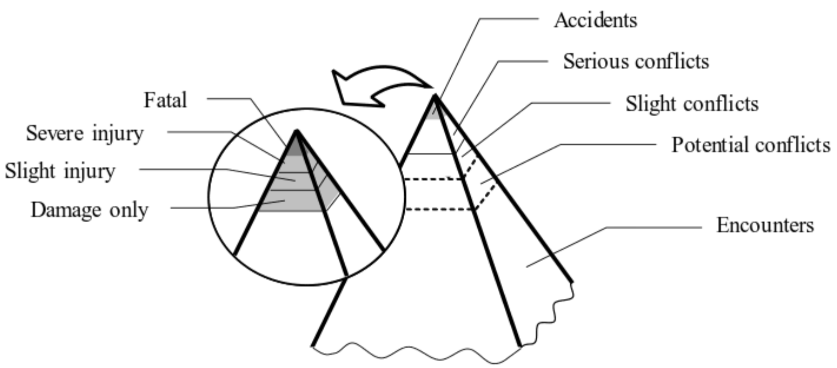


Fig. J.1: "Safety pyramid", adopted from [3].

ing used on a large scale. Previously, human observers were tasked to detect, classify and record the relevant events, all in real-time while being in the traffic environment. The high costs of using human observers, as well as some doubts in their reliability were too discouraging.

It is already a common practice in SMoS studies to use video recordings either as a complementary documentation for field observations or as a main data source. With a proper camera perspective and resolution, the measurements of road user positions and speeds taken from video can be very accurate [6]. Fully automated tools able to detect and track road users in video are already in use [7, 8]. The general concepts of such tools are illustrated in Figure J.2. The next challenge is to make the computer vision algorithms more stable while processing longer video sequences that include less favourable conditions, e.g. congested traffic, precipitation, twilight and night, while making them more practical to use.

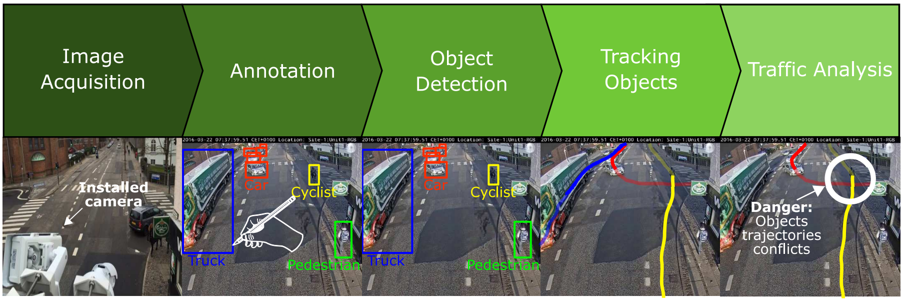


Fig. J.2: General concept for automated traffic analysis. Videos are captured via installed cameras. Humans then annotates images with bounding boxes, used to train and apply an object detector. The detected objects are tracked across time, generating trajectories which allows computing of SMoS.

Traffic safety and computer vision are two different worlds and the communication between the researchers of these two domains is not always straightforward. The following list summarizes the specific "expectations" from the traffic side that has to be taken into consideration while developing a computer vision tool:

- Majority of the indicators suggested to measure the severity of a traffic event are based on temporal and spatial proximity of the road users. Thus, the most important data to extract from video are the positions and speeds of the road users, complemented with at least rough estimate of their type and size.
- Traffic analysis requires measurements related to the road surface (e.g. speed is to be measured in meters per second rather than pixels per frame) requiring an accurate calibration model.
- Though more frequent than crashes, the events used to calculate SMoS are still relatively rare. Depending on the definition of SMoS chosen, the observation period necessary to collect a sufficient number of the relevant events might vary from 8-10 hours to several weeks.
- The observation period is limited in time, making it common to use temporary installations for the recording equipment. This put less constrains on the complexity of the video analysis algorithms as they can be processed off-line.
- Traffic environment is a public space and special rules to how the data collected should be handled apply. Ideally, pre-processing could remove sensitive information while keeping the relevant data.

Current frameworks trying to bridge the gap between traffic research and computer vision are all based on more traditional computer vision approaches [9–14]. The traditional approach involve looking for movement in the image or calculating the foreground image which depending on the scene, tells something about the moving objects or foreground objects. To classify the objects, distinctive features, e.g. width, height, color, etc. are used to separate the localized objects. The features varies a lot from object to object, so the used features chosen for classification various correspondingly, but are in this case always manually selected. A traditional machine learning algorithm will then examine the selected features and maximize the distinction between each of the object's subset of features with the purpose of classifying them. This workflow is illustrated in the upper half of Figure J.3.

Computer vision have generally seen a tremendous boost as a result of past decade's hardware improvements which have lead to a large use of the well-performing data-driven methods such as deep learning [15, 16]. Deep

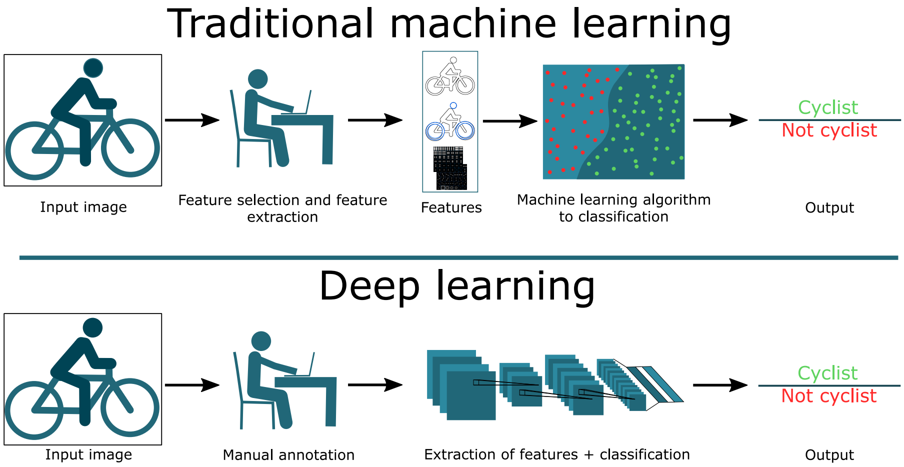

Traditional machine learning
Input image
Feature selection and feature extraction
Features
Machine learning algorithm to classification
Cyclist
Not cyclist
Output
Deep learning
Input image
Manual annotation
Extraction of features + classification
Cyclist
Not cyclist
Output

Fig. J.3: Simplified comparison of traditional machine learning approach and deep learning approach for cyclist detection. Note that manual annotations are generally faster and require less specialized knowledge than in traditional machine learning approaches.

learning is a sub-field of machine learning but differs as it does not require manually selected features. Deep learning is able to learn features that represents a given object automatically from large quantities of annotated data, which is illustrated in Figure J.2 and the lower part of Figure J.3.

In this paper, we investigate and propose a general data-driven framework to aid the cross-disciplinary communication of going from capturing video sequences and automate the traffic analysis generation using deep learning. Furthermore, we present an open-source implementation of the introduced framework.

The contributions of this paper are thus two-fold:

- Introducing and defining a data-driven cross-disciplinary framework for performing automated traffic analysis, from video acquisition to traffic analysis.
- An implementation of this framework that can detect, classify, track, and create a traffic analysis of data from an intersection.

Our implementation is released as open source and is available here: `https://github.com/ahrnbom/strudl`. This program is designed to be easy to use for traffic researchers, without extensive knowledge in computer vision.

2 RELATED WORK

In this section, we present relevant work done related to defining a framework easing collaborations between traffic researchers and computer vision researchers. The section is split into 2 parts: a part containing general established frameworks followed by relevant work where computer vision has aided traffic researchers.

2.1 General frameworks

From a computer vision perspective, several frameworks have been proposed to ease development of most general computer vision systems. In [9], a general framework is defined which is applicable for most systems working with video. The framework consists of: camera, image acquisition, pre-processing, segmentation, representation and classification. Given a set of images acquired with one or several cameras, it is possible to classify e.g. objects and actions. In [10] a video-based system for automated pedestrian conflict analysis is introduced following 5 basic components: video pre-processing, feature processing, grouping, high-level object processing and information extraction. Compared to [9], these components are more angled towards a high-level information extraction which can be considered more applicable for a traffic researcher.

In [11] a comprehensive review of computer vision techniques used for analysis in urban traffic is presented. They propose two approaches to automated traffic analysis, which differs in structure by one being a top-down approach and the other a bottom-up approach. The top-down approach estimates the foreground of the frame, e.g. frame differentiation [12]. A grouping of connected foreground pixels is done, e.g. connected component analysis, which constitutes the objects. These objects are classified [13], which can be based on heuristically predefined rules or by use of training data. Finally, tracking translate the objects into spatial-temporal domain, which provides the user with object trajectories [14]. The top-down approach analyzes the objects as a whole, whereas the bottom-up approach takes its starting point in using smaller patches of the image to detect a part of the objects, e.g. histogram of oriented gradients [17]. The detected parts of the objects are afterwards grouped together to form an object constituting the object detection step. Object detection can be extended with a classification step, where the individual object is assigned to a specific class. Finally, the objects is tracked with the purpose of creating object trajectories.

The available frameworks are in general feature-based and model-based, which have been common prior to the hardware improvements made the past decade. GPUs in particular have made training of complex artificial neural networks possible. The recent trend in computer vision is the usage of artificial neural networks, often referred to as deep learning, to do object detection by learning and adjusting the parameters and weights in the network by exposing it to large quantities of annotated data. Generally, deep learning is outperforming traditional methods by a large margin [15, 16]. The current available frameworks do not use deep learning, making our proposed framework the first to take advantage of this significant improvement in technology.

2.2 Automated video-based traffic applications

The related video-based traffic applications are split into two categories, which are object detecting and conflict-based data reduction.

Object counting

Object counting in relation to the traffic domain primarily consists of firstly detecting and classifying the object of interest, e.g. cars, trucks, pedestrians and cyclists, followed by tracking them to prevent counting the same object multiple times and to cope with potential occlusion. A lot of work has been done in especially detecting and classifying objects, in [18] they build upon the well-known Haar-like features [19], which have traditionally been used for single-frame detection. By computing such features in temporal space, the motion can estimated by comparing the absolute differences between the values in the spatial-temporal domain. In [20], object classification is done based on images captured from multiple visual traffic surveillance sensors, providing a multi-view setup which is less prone to occlusion.

As previously mentioned, the recent years of object detection has followed the hardware improvements, leading to a large use of the well-performing deep learning methods [15]. Most of the object detectors using deep learning methods, e.g. convolutional neural networks (CNN), relies on supervised learning, meaning that large quantities of annotated data is needed to train the CNN [21, 22]. In [23] a CNN was applied to the popular ImageNet Large-Scale Visual Recognition Challenge, which is a popular object recognition benchmark containing 1.2 million training images, 50,000 validation images, and 150,000 testing images. The CNN nearly halved the top-5 error rate of object recognition generated from traditional computer vision methods [16].

In general, for most of the aforementioned methods, the found objects can be tracked by using nearest neighbour, Kanade-Lucas-Tomasi feature tracker [10, 24], or by the use of more complex feature based methods, e.g. Kalman filter [25] or Hungerian tracking [26], which have proven quite useful in a wide variety of applications.

Safety analysis based on SMoS

SMoS analysis is basically a task of finding situations that fulfill certain criteria of "higher-than-normal severity" and are claimed to be a valid representation of the (un)safety problems. We omit the theoretical discussion on how such validity is established and concentrate on the practical realization of SMoS tools once the definitions are agreed upon.

Unlike the "old-fashion" traffic conflict techniques in which the human judgements of severity played an important role, implementation of SMoS in a computer vision tool requires strict mathematical definitions of what indicators and thresholds are to be used.

Several fully-automated tools with focus on SMoS analysis have been presented [27–29]. The common denominator in all of them is that the computer vision tools attempts to produce the trajectories of all road users, calculates the indicator values and relate them to the pre-defined thresholds, and finally reports the total number of conflicts detected.

One concern that can be raised, however, is about the potential overrelying on the automation. First, the aggregated numbers of conflict counts hide in themselves the errors related to the noise in the trajectory data, often acknowledged but rarely controlled for. What is more important, the complete exclusion of a human from the process results in difficulties in interpreting the results and getting inspiration for the counter-measures [30].

An alternative approach is to split the analysis in two steps - initial detection of the situations that are potentially relevant and then final filtering among them with human involvement. The validity of SMoS is increasing as the thresholds are set more strict for the high-severity events only [31]. The more severe the events are, the less is the frequency, though. This has two important practical implications: i) longer observation periods are required, which make the automation of the first task crucial; ii) not many events are expected to be found, which allows manual or semi-automated processing of the events with humans involved feasible. [2] presents an example of such study. The large volume of video data is processed using a watchdog software [32] that detects potentially relevant events. Later, the data is processed using a semi-automated tool [33] that allows an operator to extract the trajectories and speed profiles of road users with very high accuracy [6].

The proposed framework is primarily developed with the two-step approach in mind. Even though it does perform tracking of all road users and potentially can be used for fully-automated SMoS analysis, it has been tested on the task of detection of "potential conflicts" that still require post-processing.

3 FRAMEWORK OVERVIEW

In order for a framework for traffic surveillance to be useful in practice, it needs to be general enough to handle different kinds of queries and criteria. A single cross-disciplinary computer vision framework can work for multiple applications, as the main steps that need to be performed are typically the same. The proposed framework in this paper takes its spawn in a top-down approach, as presented in [11], and some of the general concepts presented in [9, 10]. In Figure J.4, the proposed framework is illustrated in a block flow diagram. Each block forms the structure for the following of this section and will thus be described accordingly.

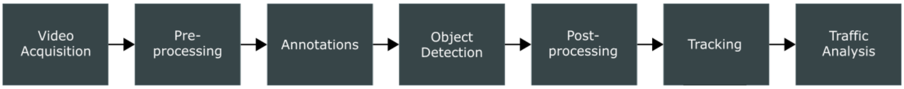


Fig. J.4: The proposed framework for automated traffic analysis. While Video Acquisition and Traffic Analysis can be considered to belong to the field of traffic research, the remaining central blocks belong to the field of computer vision.

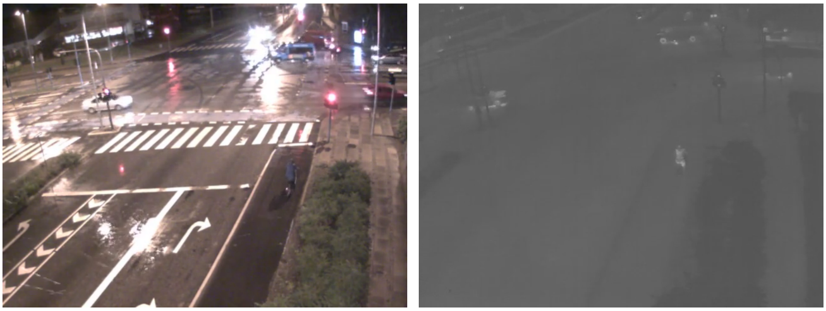
3.1 Video Acquisition

The first step in the general framework is video acquisition. The primary goal is to acquire video data to the pipeline. Essential considerations to do are presented in the following subsection.

Modalities

The most common sensor for acquiring video data is a traditional RGB camera, which is similar to the human eye making the videos easy to interpret. Other options include using a thermal camera, which during the last decade has seen a price reduction making it feasible to use in traffic surveillance applications [34]. A thermal camera is a passive sensor that captures the infrared radiation emitted by all objects, which can be translated to "seeing" the temperature. The thermal camera are thus usable in the night which can be an advantage compared to RGB, but can also be considered a disadvantage as the lack of color information can make classification challenging as seen in Figure J.5.

The choice of modality, e.g. RGB or thermal, does not affect the rest of the suggested framework, the choice comes down to a matter of cost, expected light, weather conditions and privacy concerns. Specifications of the sensor should be taken into consideration, e.g. frames per second (FPS) and resolution.



(a) RGB Camera. **(b)** Thermal Camera.

Fig. J.5: Example of the RGB modality and thermal modality captured at a traffic intersection simultaneously during a rainy night.

Camera calibration

By carefully measuring the positions of some points visible in the camera, the camera can be calibrated, allowing positions in pixel coordinates in the images to be translated to world coordinates. If this step is omitted, any results found by computer vision algorithms are significantly more difficult to interpret and use since they cannot be converted to world coordinates. Detailed search queries and SMoS typically need to be computed in world coordinates to be useful.

3.2 Pre-processing

Modern object detectors using CNNs do not need much pre-processing. The only form of pre-processing used in our framework is masking. Often, the entire scene captured by the camera is not of interest; if an application is to find interesting situations in a crossing, then it is of no importance what happens far from that crossing. For these cases, a manually drawn "do-not-care" zone is created as an overlaying mask or the image is simply cropped. This speeds up annotations and helps training a reliable object detector. If this step is omitted, and only parts of the images are annotated, this may confuse the detector during training, possibly leading to reduced accuracy.

3.3 Annotations

CNNs learn by examples, so a human needs to define and annotate this example many times before the network can be trained to do the same. In this pipeline, neural networks are used for object detection only, so the annotations consist entirely of marking objects in images, by bounding boxes and assigning a class label to each box, as illustrated in Figure J.6.

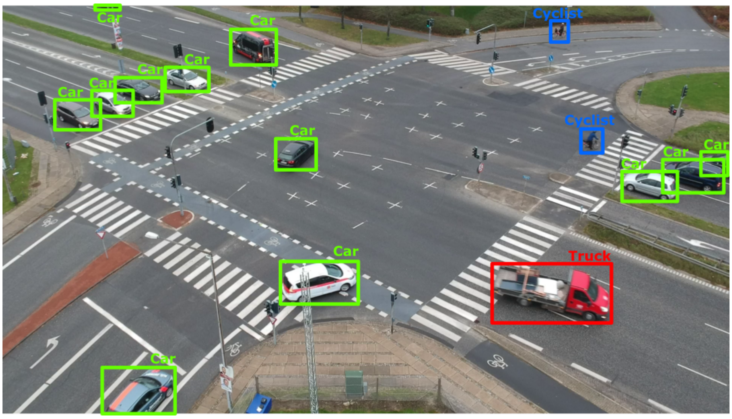


Fig. J.6: Bounding box annotations with belonging class label.

It is important that all visible and relevant objects are annotated. Otherwise, these will be considered negative examples when the detector is trained. For example, if one car is marked as a "car" and another one is not, then the detector will have a hard time understanding why one is considered a "car" while the other one is not, and accuracy may suffer as a result.

If a large dataset of traffic images were annotated and made publicly available, they could be used if their viewing angle, lightning conditions etc. are reasonably similar to the new data. In such a case, smaller amounts of new data are needed to be annotated. Despite the current lack of such a dataset, pre-training the networks on general images reduces the number of image annotations needed to a couple of hundred, as opposed to thousands or more.

3.4 Object Detection

The goal of object detection is to find "objects", e.g. road users, as axis-aligned bounding boxes with class labels in an image. Traditionally this step has been split into two steps: localization (finding the bounding box) and classification (assigning a class to the bounding box), but due to recent years' advancements in CNN designs, both can be performed in a single step. The choice of class labels is application dependent and is not limited to a specific amount. Multiple can be used, if they are of particular interest, but it should be noted that a significant amount of examples has to appear in the annotated images for the detector to become accurate.

3.5 Post-processing

In the post-processing step, the movement direction for each of the detected objects can be computed and converted to world coordinates.The projection to world coordinates can be done using TSAI camera calibration. The model is calibrated based on a set of points measured both in image coordinates and in the road plane using special equipment [6]. Movement directions are useful cues when connecting the detections into tracks. Performing the tracking in world coordinates has benefits, mainly being more independent of the viewing angle, and working directly in natural units and world coordinates allows more detailed and natural track analysis.

3.6 Tracking

Tracking consists of connecting the detected objects in spatio-temporal space, meaning that each detected object in the video needs to be either associated with a previously existing track or as a completely new track in the video. Though this might sound as a somewhat easy task, several challenges are introduced when objects radically change direction or if multiple objects get too close to each other in the sensor's field-of-view.

The performance of the object detection is critical for proper tracking as trajectories cannot be generated for objects that are not detected. Tracking can, however, compensate for some issues in the detector. For example, if a vehicle is detected in only 1 frame, but not in any of prior or following frames, there is a high probability that this is a false detection. If an object is not detected in a small number of frames, but is detected before and after, the tracking algorithm may be able to understand that it is indeed the same object.

Selecting a sensor with too low FPS results in objects in the scene moving a large distance between the consecutive captured frames, which can make it harder to connect the detected objects in spatio-temporal space. Using a high FPS, the objects' movement between consecutive captured frames becomes less, which generally makes tracking easier. Videos with 15 FPS seem to work in our experiments.

3.7 Traffic Analysis

The final step of the proposed framework is to analyze the road user tracks with respect to safety. For example, indicators like Time-to-Collision and Post-Encroachment Time can be calculated and events with severity above a certain threshold can be detected and presented to the user for further examination. The data about the distribution of events within different severity categories can be used by special statistical methods such as extreme-value theory in order to estimate the expected number of crashes [35, 36]. Also, trajectory data can be used for calculations of advanced exposure measures, for example a number of encounters between road users of a certain type and performing a certain manoeuvre [37]. Clustering of the trajectories and detection of deviant trajectories do not fit into any of clusters may reveal the abnormal incidents such as movement in wrong direction or stop at an unusual place.

Since the tracks are computed in world coordinates, thresholds, safety measures and other criteria can be expressed in natural terms and units. While traditional computer vision systems allow only simple criteria (typically expressed in pixel coordinates), world coordinate tracks allow for arbitrarily complex queries, that are more meaningful from a traffic analysis perspective.

4 EXPERIMENTS

As a part of a traffic analysis project, an intersection with a crossing of interest in Malmö, Sweden was filmed for 24 hours with a thermal camera. TSAI calibration [38] was computed by measuring 57 points visible in the videos. People were hired to watch through the entire 24 hours of video, tasked to find times in the video where both a car and either a pedestrian or a bicyclist are visible at the same time, where the car will at some point make a turn to pass the crossing of interest, while the pedestrian/bicyclist will at some point pass the crossing. See Figure J.7 for a visual explanation of the task. These times were then inspected in more detail by traffic analysts. We stress that this is not a "toy problem"; the human watchers were required as a starting point for further traffic research at this intersection, and we hope that the existence of this framework can reduce the need for human labor in situations like these in the future.

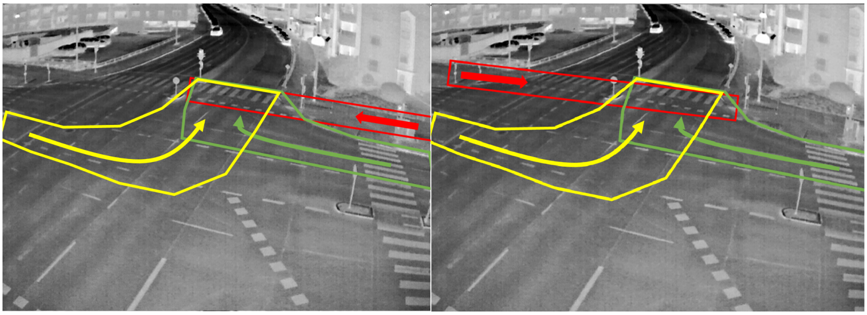


Fig. J.7: The goal is to find times when a vulnerable road user is moving through the red regions in the marked directions, while a car is moving either through the green or yellow regions simultaneously.

An implementation of the suggested framework was used to perform the same task, using the human observer's results as ground-truth. As a baseline, the Road User Behaviour Analysis (RUBA) software [32], which is a traffic analysis tool based on traditional computer vision technology, was also tested for the same task.

There is some ambiguity in when exactly during an encounter it is detected by an observer or computer vision tools. Therefore, it was allowed for some time discrepancy for a detection to be counted as correct. By testing multiple time distance thresholds between the ground truth and the output of the automatic systems, a trade-off between precision and recall can be observed. We use precision and recall curves to visualize this trade-off and compare the automatic systems.

4.1 STRUDL: description of implementation

This section describes how our framework following the definitions in Section 3 was implemented, in order to solve the problem described above. The implementation is called **S**urveillance **Tr**acking **U**sing **D**eep **L**earning (STRUDL). It can be used in any context where objects seen from a static camera need to be tracked. Those tracks can be analyzed to for example find times of interest. While thermal videos were used in this experiment due to privacy concerns, the STRUDL system works with RGB as well (and should in fact perform better with RGB as the pre-training of the object detector is made with RGB images). The remaining parts of this section will describe in more detail how STRUDL implements the computer vision parts of the suggested framework.

Pre-processing

With modern object detection algorithms based on CNN, very little pre-processing of images is necessary. The only pre-processing done is applying a visual "do-not-care" mask.

Annotation

500 frames were selected from the collected videos and annotated manually with bounding boxes and class labels. The frames were taken from 25 randomly selected 5 minute clips, and from each such clip, 20 frames were sampled evenly. This way, there should be a large variety in the road users appearing in the images. A variant of Extreme Clicking [39] was implemented to make the annotation process fast. The reason why 500 frames is sufficient to get decent object detection performance is that the detector is pre-trained on a general objects detection task. Training the object detector from scratch would require drastically many more images.

Object Detection

The object detector SSD [40] was used. It is a commonly used CNN for the object detection task for its reasonable trade-off between accuracy and execution speed. On a powerful modern GPU, it runs in around real-time. The objects found are presented as axis-aligned bounding boxes. The SSD network was pre-trained on the large MS COCO dataset [41], which contains a large amount of images with bounding box annotations of many different kinds of objects (not only traffic-related ones), made by human annotators. Then, the network was fine-tuned on images from the videos for which the experiment is conducted, as described in Section 4.1. Finally, the object detector is applied to every single image, and detected objects are stored.

Post-processing

For the videos, the OpenCV function `goodFeaturesToTrack` was used to find points which can be tracked, and then by repeatedly using the OpenCV function `calcOpticalFlowPyrLK` [42], those points were turned into point tracks. These tend to follow how objects move in the scene. For each detected bounding box, the average movement direction of point tracks moving through the box were computed, giving each box a movement direction.

Then, using a TSAI camera calibration model [6, 38], each such box and movement direction were converted to world coordinates. Because of the pixel-aligned nature of bounding boxes, only the center point was converted. Because the orientation of road users can be computed from their movements directions, and the class labels allow approximate 3D models to be inserted in their place, any information about the movements, position and spatial extent of the road users should be possible to obtain, at least approximately, from this simple representation.

Tracking

A simple Hungarian tracker [26] was used, using class consistency, position in world coordinates and movement direction to compute the association cost. World coordinate detections that were not associated to any existing tracks, were made into tracks of their own unless they were too close to some already existing track. When no detection were associated with a given track, the track continues along its previous direction for some time until being removed, unless it is associated with a new detection before that. Tracks that were short-lived, that were only associated with one or two detections were removed, as they are often false or unreliable tracks.

The tracking requires tuning of 13 parameters, which were optimized using a blackbox optimization scheme for a short video clip (15 seconds long) where ground truth tracks in world coordinates were created for each road user, which took around 30-40 minutes of human labor to create. Because the tracks are in world coordinates, it is believed to be possible to re-use the optimized parameters for a different viewpoint, perhaps with minor changes.

Traffic Analysis

The goal was to find times when at least two tracks are visible at the same time while the two tracks intersects at some point, e.g. car move to turn and cross the vulnerable road user track. To implement this as a traffic analysis program, mask images were drawn which mark the interesting regions, and the tracks were tested to see if they at some points move through the marked regions. The mask images can be seen in Figure J.8.

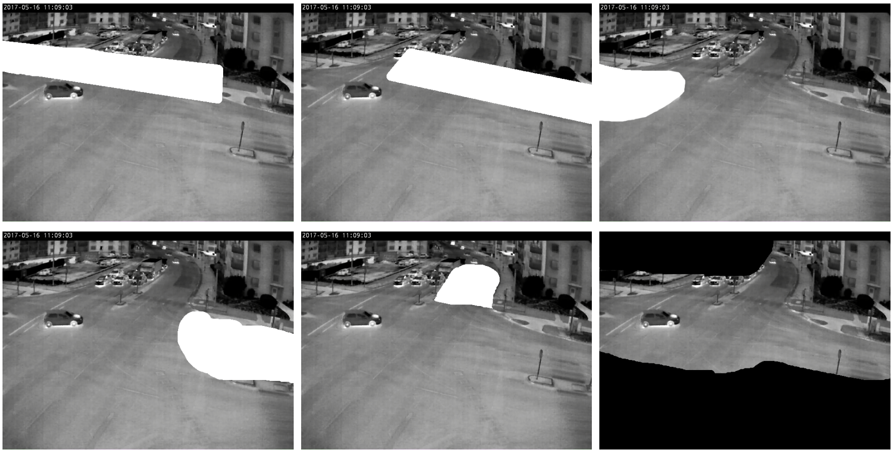


Fig. J.8: Checking masks used for the experiment. Top left: VRUs moving to the right. Top center: VRUs moving to the left. Top right: first required position of left-turning cars. Bottom left: first required position for right-turning cars. Bottom center: the last required position of both left-turning and right-turning cars. Bottom right: mask used during object detection annotations, in order to save annotation time. The same mask is used when running the object detector.

4.2 Results

The results are seen in Figure J.9, where the proposed system is compared to RUBA [32]. RUBA's raw output was compared directly, and after seeing that the number of false positives were very high (leading to a low precision), time was spent to remove 967 of RUBA's found situations by manually examination ("RUBA+human" in the figure). Most of all the removed events were indeed false positives, as the recall drops very little in this process. Even so, the number of false positives remain high for left-turning cars. The manual time spent with RUBA was around three hours, where around 90 minutes were spent manually removing false positives. Our system, on the other hand, required only around two hours of manual work constructing the detection annotations, and around 30-40 minutes spent on tracking ground truth. Also note that the human time can decrease as the software becomes more used, allowing training images from similar viewing angles to be reused, and tracking parameters might be possible to transfer with little to no changes, because they are expressed in world coordinates. Furthermore, for a human to make annotations, little training is required, while designing hitboxes and thresholds for RUBA requires experience and familiarity with the software.

It should be noted that the problem was significantly more difficult for left-turning cars than for right-turning cars. The exact cause for this is not yet known. Only 10 situations with left-turning cars were marked as interesting by the human annotators, compared to 331 for right-turning cars during this one day of video.

We stress that this comparison between STRUDL and RUBA does not include the main difference between the two; while RUBA provides only "take-it-or-leave-it" times of interest, STRUDL provides full tracks in world coordinates that can be further analyzed, by e.g. computing SMoS, sorting by severity or further filtered.

Some tracking example results can be seen in Figure J.10. The tracking generally works well, but there is also some room for improvement in its robustness for some tracks.

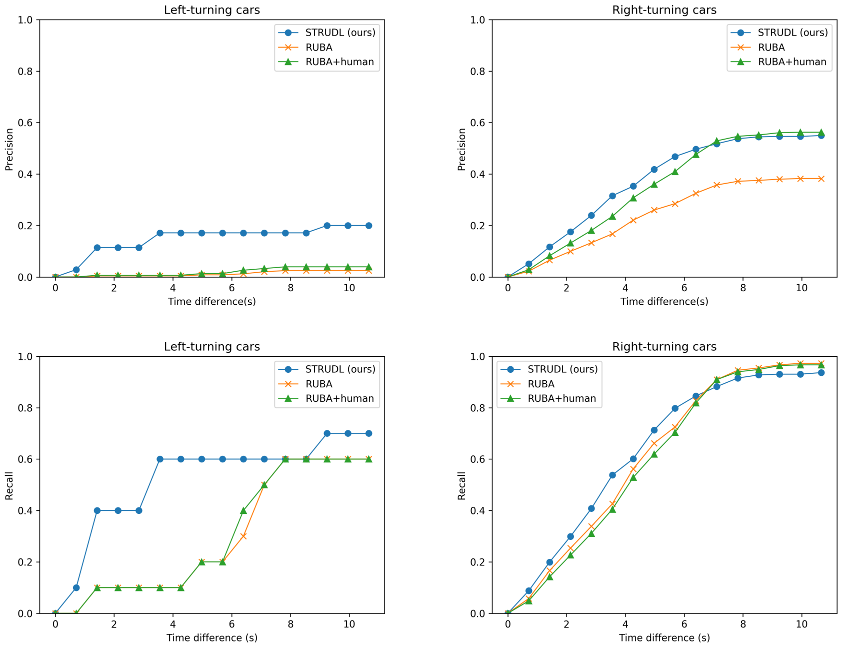


Fig. J.9: Precision and recall for the experiment, against the time difference for which a detected time can differ from the ground truth time and still be considered correct. RUBA+human reaches STRUDL's precision for right-turning cars, while STRUDL is still better for left-turning cars. For recall, they perform similarly for right-turning cars, and are able to find more than 95% of the ground truth times within ±10 s, while for left-turning cars, STRUDL's recall is clearly better for short time differences, while being only slightly better for longer time differences.

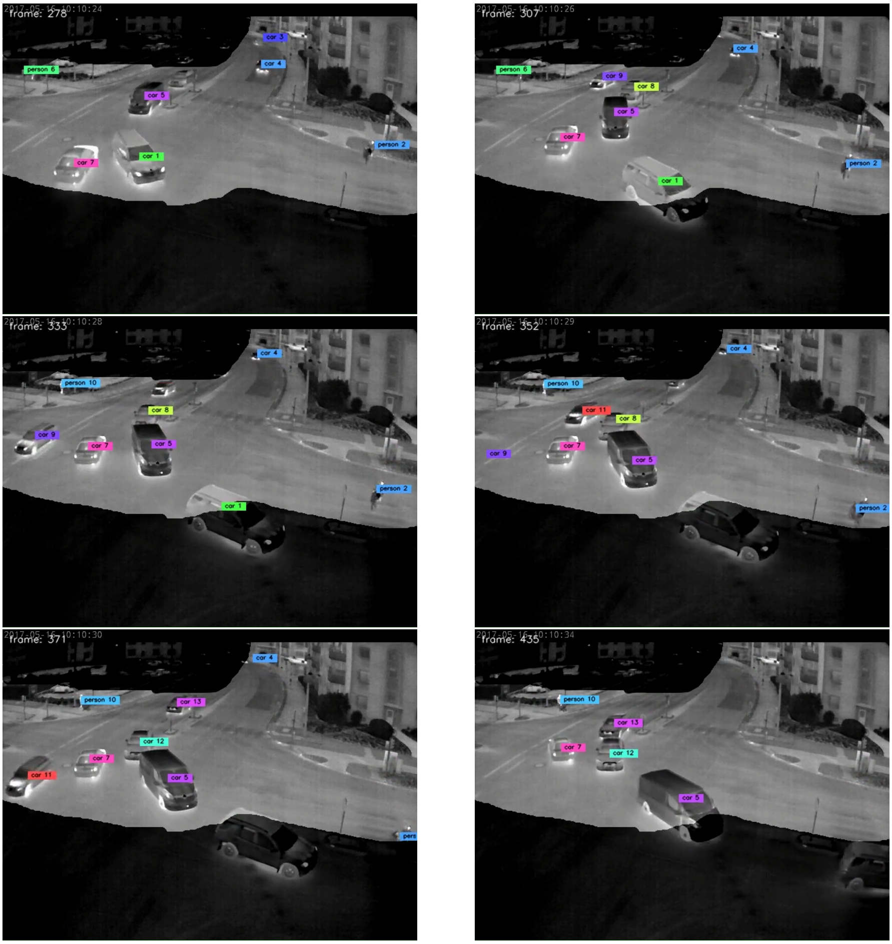


Fig. J.10: Example of tracking results from our experiments. An example of bad tracking is the person walking with a stroller, who is broken up into two tracks. "Car 9" is lost and it takes a few frames for the tracking algorithm to remove this track. For the most part however, tracking works as expected. The dark areas are the masked "do-not-care" zones. Best viewed in color.

The experimental results show that the proposed framework works, and the STRUDL implementation is better than traditional approaches for tasks of this kind. It is flexible, meaning that if one is dissatisfied with the results for a given problem, the path forward is often clear. If the object detector makes too many mistakes, more training data is needed. If the tracking fails too often, the parameters can be tuned, manually or via data-driven optimization. If there are too many false positives, the analysis criteria can be modified with relatively little effort. Visualizations of the different steps of the computer vision pipeline make it easy to pinpoint where issues arise.

More importantly, where traditional computer vision system have a limited range of possible operations, the richness of full trajectories allow for much more freedom. It is possible to compute SMoS or other measures of interest, to filter or sort the detected situations by severity. The proposed system can therefore be seen as a starting point for arbitrarily complex traffic analysis, whereas traditional methods are essentially of a take-it-or-leave-it nature, impossible or difficult to further analyze, filter, sort and work with.

The proposed automatic system needs some human assistance, mainly in annotating image data to train the object detector. When looking at new video data, the amount of new annotations necessary will depend heavily on previously available data. Manually annotating some images seem like a good trade-off, as opposed to traditional methods requiring time-consuming parameter tuning, as it is relatively simple and fast, and if multiple somewhat similar views are studied, annotations from one view can be re-used, reducing the annotation time per intersection.

One limitation of the proposed framework is the tracking algorithm, which is quite simple in nature. It is known to sometimes make mistakes when tracks get too close to each other, or if the detector fails to locate an object for many frames. These flaws could possibly be fixed or reduced by letting a neural network perform the tracking, but that would require a large amount of annotated ground-truth tracks for training which take time to produce. Our implementation requires little to no annotated ground-truth tracks, since tracking parameters should be mostly transferable between views. Still, it would be of interest to test and compare different tracking algorithms for this setting. The modular implementation of STRUDL makes it relatively simple to replace the current tracking algorithm, should so be needed. Another limitation is the lack of uncertainty measures in the STRUDL software. There is no universally accepted standard for how to measure the certainty of detections and tracks, but some combination of detection confidence and the similarities between every track and typical trajectories could probably be used for this purpose. This is one promising direction for future work.

The implementation of STRUDL is designed with flexibility in mind and it is our intention to continually improve the software. For example, it would be useful to have built-in support for computing SMoS, or make improvements to its computer vision algorithms, tracking in particular. We also hope that other implementations of the proposed framework will arise, to suit the specific needs of different traffic analysis problems.

5 CONCLUSION

We present the, to the best of our knowledge, first cross-disciplinary framework for automated traffic surveillance analysis to take advantage recent improvements in data-driven deep-learning. Through experiments with our open-source implementation, STRUDL, we show better results than traditional systems, while opening new possibilities by providing full trajectories in world coordinates, allowing arbitrarily complex traffic analysis. Promising future works includes computing certainty measures and SMoS automatically and improving the stability of tracking.

6 ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] European Commission, *2017 road safety statistics: What is behind the figures? - Fact Sheet*. European Commission, 2018. [Online]. Available: http://europa.eu/rapid/press-release_MEMO-18-2762_en.pdf

[2] A. Laureshyn, C. Johnsson, T. Madsen, A. Várhelyi, M. de Goede, A. Svensson, N. Saunier, and W. van Haperen, "Exploration of a method to validate surrogate safety measures with a focus on vulnerable road users," *International Conference on Road Safety and Simulation*, 2017.

[3] C. Hydén, "The development of a method for traffic safety evaluation: the swedish traffic conflict technique," *Department of Traffic Planning and Engineering. Bulletin 70*, vol. Doctoral thesis. Lund University, 1987.

[4] J. H. Kraay, "Proceedings of the third international orkshop on traffic conflicts techniques," *SWOV, R-82-27*, Apr 1982.

[5] S. J. Older and J. Shippey, "Proceedings of the second international traffic conflicts technique workshop," *Transport and Road Research laboratory*, May 1980.

[6] A. Laureshyn and M. Nilsson, "How accurately can we measure from video? practical considerations and enhancements of the camera calibration procedure," *Transportation Research Record*, p. 0361198118774194, 2018.

[7] S. Knake-Langhorst, K. Gimm, T. Frankiewicz, and F. Köster, "Test site aim – toolbox and enabler for applied research and development in traffic and mobility," vol. Transportation Research Procedia 14, pp. 2197–2206, 2016. [Online]. Available: https://doi.org/10.1016/j.trpro.2016.05.235

[8] K. Ismail, T. Sayed, and N. Saunier, "Automated safety analysis using video sensors: technology and case studies," vol. Canadian Multidisciplinary Road Safety Conference, 2010.

[9] T. Moeslund, *Introduction to video and image processing: Building real systems and applications*. Springer, 2012.

[10] K. Ismail, T. Sayed, N. Saunier, and C. Lim, "Automated analysis of pedestrian-vehicle conflicts using video data," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2140, pp. 44–54, 2009.

[11] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, Sept 2011.

[12] S.-C. S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 14, p. 726261, 2005.

[13] N. Buch, J. Orwell, and S. A. Velastin, "Urban road user detection and classification using 3d wire frame models," *IET Computer Vision*, vol. 4, no. 2, pp. 105–116, 2010.

[14] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE PAMI*, vol. 30, no. 10, pp. 1683–1698, 2008.

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[18] M. J. Jones and D. Snow, "Pedestrian detection using boosted features over many frames," in *ICPR*. IEEE, 2008, pp. 1–4.

[19] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance." IEEE International Conference on Computer Vision, 2003, p. 734.

[20] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE*, vol. 5, pp. 24 417–24 425, 2017.

[21] D. Oñoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *ECCV*. Cham: Springer, 2016, pp. 615–629.

[22] M. Ahrnbom, M. B. Jensen, K. Åström, M. Nilsson, H. Ardö, and T. Moeslund, "Improving a real-time object detector with compact temporal information," in *IEEE International Conference on Computer Vision Workshops*, Oct 2017, pp. 190–197.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[24] C. Tomasi and T. Kanade, "Detection and tracking of point features," International Journal of Computer Vision, Tech. Rep., 1991.

[25] A. Hanif, A. B. Mansoor, and A. S. Imran, "Performance analysis of vehicle detection techniques: A concise survey," in *Trends and Advances in Information Systems and Technologies*. Cham: Springer, 2018, pp. 491–500.

[26] F. Bourgeois and J.-C. Lassalle, "An extension of the munkres algorithm for the assignment problem to rectangular matrices." vol. 14, pp. 802–804, 12 1971.

[27] A. Tageldin, T. Sayed, and K. Ismail, "Evaluating the safety and operational impacts of left-turn bay extension at signalized intersections using automated video analysis," *Accident Analysis and Prevention*, vol. 120, pp. 13 – 27, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457518303385

[28] M. G. Mohamed and N. Saunier, "The impact of motion prediction methods on surrogate safety analysis: A case study of left-turn and opposite-direction interactions at a signalized intersection in montreal," *Journal of Transportation Safety & Security*, vol. 10, no. 4, pp. 265–287, 2018. [Online]. Available: https://doi.org/10.1080/19439962.2016.1255690

[29] Å. Svensson, A. Laureshyn, T. Jonsson, H. Ardö, and A. Persson, "Collection of micro-level safety and efficiency indicators with automated video analysis," in *3rd International Conference on Road Safety and Simulation, Indianapolis, Ind*, 2011.

[30] A. Laureshyn, M. de Goede, N. Saunier, and A. Fyhri, "Cross-comparison of three surrogate safety methods to diagnose cyclist safety problems at intersections in norway," *Accident Analysis and Prevention*, vol. 105, pp. 11–20, 2017.

[31] C. Johnsson, A. Laureshyn, C. D'Agostino, and H. Farah, "Surrogate measures of safety," in *InDeV, Horizon 2020 project. Deliverable 3.1.*, 2018.

[32] T. K. O. Madsen, C. H. Bahnsen, M. B. Jensen, H. S. Lahrmann, and T. B. Moeslund, "Watchdog system," in *InDeV, Horizon 2020 project. Deliverable 4.1.*, 2016.

[33] C. Johnsson, A. Laureshyn, and H. Norén, "T-analyst - semi-automated tool for traffic conflict anlysis," in *InDeV, Horizon 2020 project. Deliverable 6.1.*, 2018.

[34] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine Vision and Applications*, vol. 25, no. 1, pp. 245–262, Jan 2014. [Online]. Available: https://doi.org/10.1007/s00138-013-0570-5

[35] P. Songchitruksa and A. P. Tarko, "The extreme value theory approach to safety estimation," *Accident Analysis & Prevention*, vol. 38, no. 4, pp. 811–822, 2006.

[36] A. P. Tarko, "Surrogate measures of safety," in *Safe Mobility: Challenges, Methodology and Solutions*. Emerald Publishing Limited, 2018, pp. 383–405.

[37] R. Elvik, "Some implications of an event-based definition of exposure to the risk of road accident," *Accident Analysis & Prevention*, vol. 76, pp. 15 – 24, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000145751400390X

[38] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," 01 1986.

[39] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," *CoRR*, vol. abs/1708.02750, 2017. [Online]. Available: http://arxiv.org/abs/1708.02750

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[41] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[42] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," vol. 1, 01 2000.

Part V

Knowledge for the World

Paper K

Deep Learning - et gennembrud indenfor kunstig intelligens

Morten B. Jensen, Chris H. Bahnsen, Kamal Nasrollahi and Thomas B. Moeslund

The paper has been published in the
Aktuel Naturvidenskab Vol. 2, pp. 8–13, 2018.

The layout has been revised.

I løbet af de seneste 10 år er kunstige, neurale netværk gået fra at være en støvet, udstødt teknologi til at spille en hovedrolle i udviklingen af kunstig intelligens. Dette fænomen kaldes deep learning og er inspireret af hjernens opbygning.

1 INTRODUCTION

Hvordan kan en computer vinde over verdensme- steren i GO, hvor der er flere mulige kombinationer på spillepladen end atomer i universet? Hvordan kan en bil forstå, at der er en fodgænger foran den og selv bremse?

Svaret på denne type spørgsmål er intelligente computersystemer, der lærer ved at analysere data – rigtig meget data. Den nyeste metode indenfor dette forskningsområde kaldes deep learning. Metoden har på få år revolutioneret store dele af forskningsverdenen og er nu på vej ud i alle grene af samfundet, hvor den forventes at få afgørende betydning.

Et gammelt ordsprog siger, at viden er magt! Måske er data et af de vigtigste elementer i dannelsen af viden, men hvordan man styrer og udnytter data, er endnu vigtigere. Derfor har forskere altid forsøgt at udvikle avancerede måder til indsamling af data for derefter at udnytte det bedst muligt. For at finde inspiration til at udvikle bedre databehandlingsteknikker har forskere kigget på hjernens opbygning og opførsel i håb om at kunne opnå en forståelse, der efterfølgende kan implementeres i computere. Dette forskningsområde kendes også som kunstig intelligens. På grund af hjernens komplekse struktur har det altid været meget udfordrende at forstå hjernens grundlæggende funktionalitet for derefter at opbygge et hjerne-lignende system. På trods af, at ingeniører længe har formået at konstruere systemer, der kan ef- terligne hjernen ved simple opgaver, så har forskere stødt hovedet mod muren, når det kom til at konstruere systemer, der er i stand til at løse mere udfordrende opgaver, for eksempel genkendelse af objekter.

Imidlertid har nylige fremskridt inden for dataindsamling og rå processeringskraft gjort det muligt at bygge systemer baseret på kunstig intelligens, der kan løse komplekse problemer som objektdetektion, genkendelse og tracking. Systemerne er nu så gode, at de i nogle tilfælde klarer sig bedre end menneskelige eksperter.

Disse systemer bliver trænet ved hjælp af massive datamængder gennem matematiske algoritmer, der er bedre kendt under paraplybetegnelsen deep learning. Før vi kommer nærmere ind på det, må vi en tur omkring hjernen for at få en grundlæggende forståelse af disse systemer.

2 HJERNEN

Hjernen er en af de mest komplekse strukturer, vi kender. Den er opbygget af 100 milliarder celler kaldet neuroner, og der er cirka samme antal neuroner i hjernen, som der er stjerner i Mælkevejen. I figur K.1 ses en illustration af et neuron. Hvert neuron har: Et cellelegeme, indeholdende kernen, som er neuronets behandlingscenter. Et sæt indgangsforbindelser, dendritter, som bringer signaler fra de andre neuroner til kernen i det nuværende neuron. En axon, som overfører resultaterne af behandlingen af indgangssignalerne i kernen til de neuroner, der er forbundet til det aktuelle neuron via sine udgangsforbindelser (axonterminaler).

En gruppe af disse små hjerneneuroner, der er internt forbundet med hinanden, er ansvarlige for at udføre en specifik opgave. For eksempel udføres matematiske operationer i en bestemt del af hjernen, mens følelser opfattes af en anden gruppe neuroner. Ved løsning af specifikke opgaver viser de ansvarlige grupper af neuroner mere elektrisk aktivitet end resten af hjernen. Disse elektriske aktiviteter skyldes frigivelse af kemiske stoffer mellem neuronerne, der er internt forbundet med hinanden. Hvis summen af kemiske stoffer ved neuronet er større end et bestemt niveau, bliver neuronet aktiveret. I modsat fald forbliver det passivt.

Når vi som menneske prøver at lære en bestemt opgave, for eksempel når en baby lærer at gå, gennemføres denne læring gennem adskillige forsøg. Under disse forsøg lærer hjernen, eller rettere: En specifik gruppe neuroner lærer, hvordan de skal aktiveres for at udføre den specifikke opgave. Mæng-

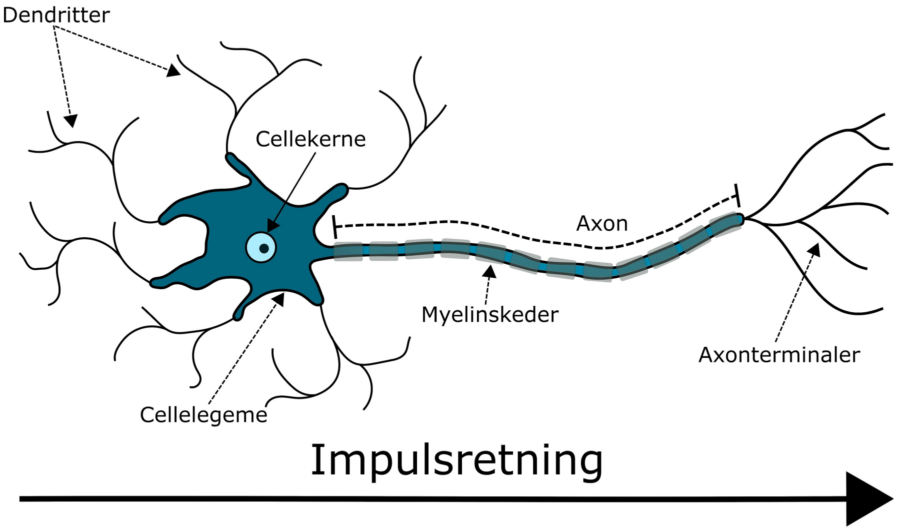
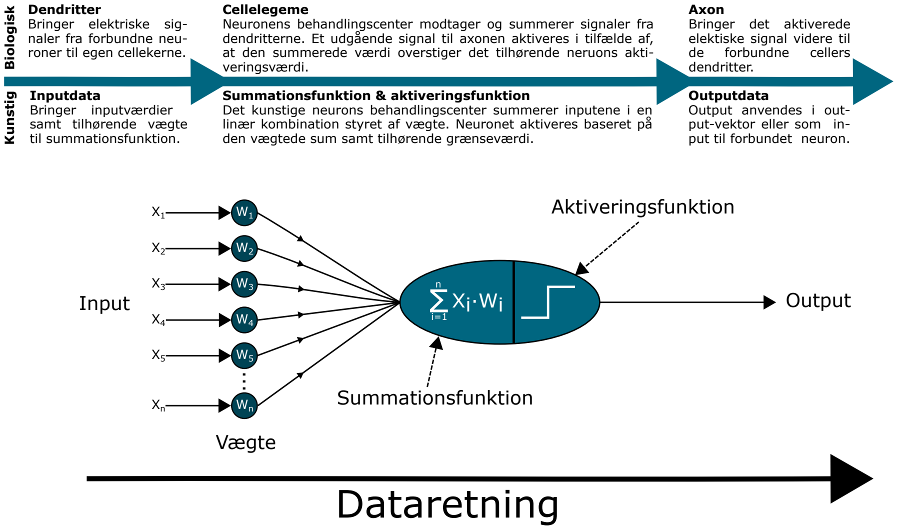


Fig. K.1: Illustration af et neuron, som er hjernens byggesten.

den af de kemiske stoffer, der frigives mellem neuronerne, definerer graden af forbindelse, også kaldet vægtningen, mellem de tilsluttede neuroner.



Dendritter
Bringer elektriske signaler fra forbundne neuroner til egen cellekerne.

Cellelegeme
Neuronens behandlingscenter modtager og summerer signaler fra dendritterne. Et udgående signal til axonen aktiveres i tilfælde af, at den summerede værdi overstiger det tilhørende neruons aktiveringsværdi.

Axon
Bringer det aktiverede elektiske signal videre til de forbundne cellers dendritter.

Inputdata
Bringer inputværdier samt tilhørende vægte til summationsfunktion.

Summationsfunktion & aktiveringsfunktion
Det kunstige neurons behandlingscenter summerer inputene i en linær kombination styret af vægte. Neuronet aktiveres i tilfælde af, at den vægtede sum samt tilhørende grænseværdi.

Outputdata
Output anvendes i output-vektor eller som input til forbundet neuron.

Fig. K.2: Princippet i en kunstig neuron.

Man kan simulere det biologiske neuron med en matematisk funktion, der består af en lineær kombination af alle inputs til neuronet. Den lineære kombination styres af vægtene af de enkelte inputs. Denne sum af vægtede input svarer til mængden af de kemiske stoffer, der kommer til et neuron. Derefter bestemmer en såkaldt aktiveringsfunktion, om neuronet skal aktiveres eller forblive passivt. Hvis den vægtede sum er større end en given grænseværdi, aktiveres neuronet. Dette princip er illustreret i figur K.2.

Et kunstigt neuralt netværk består af en kombination af disse kunstige neuroner i forskellige lag, der er internt forbundet med hinanden gennem vægtede forbindelser. Antallet af lag beskriver dybden af netværket. Man betegner et neuralt netværk som dybt, hvis det indeholdertre eller flere lag.

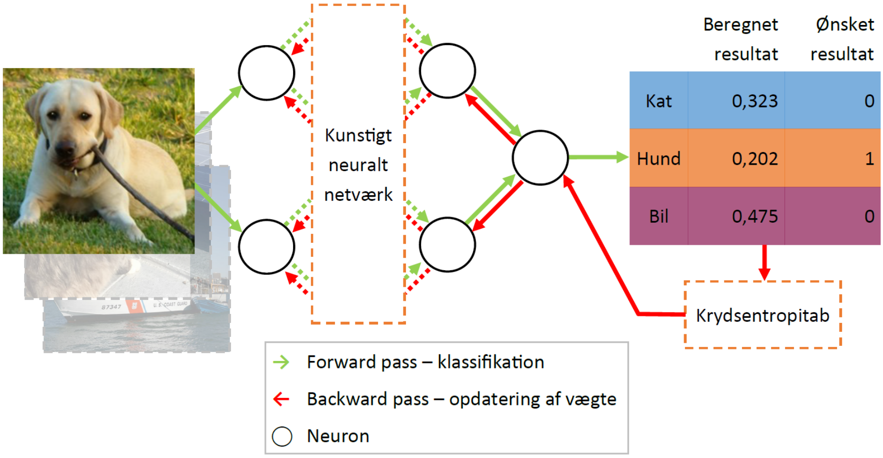
Aktiveringsfunktioner spiller en nøglerolle i kunstige neurale netværk. Hvis aktiveringsfunktionen udelukkende består af lineære funktioner, kan det kunstige neurale netværk udelukkende beskrive lineære fænomener, og dets samlede funktion kan grundlæggende beskrives af én stor matrix. Hvis aktiveringsfunktionen derimod ikke kan beskrives som en lineær kombination af dens input, er det kun dybden af det kunstige neurale netværk, der begrænser kompleksiteten af de funktioner, som netværket kan beskrive.

3 LÆRING

For at lære et kunstigt neuralt netværk at udføre en specifik opgave kræves en læringsalgoritme, hvis formål er at finde de rette vægte mellem netværkets neuroner. Vægtene læres gennem adskillige iterationer, hvor det neurale netværk præsenteres for store mængder træningsdata. Hver enkelt stykke data er annoteret, det vil sige, at det er parret med den ønskede respons fra det neurale netværk – for eksempel at et billede af en hund hører til kategorien "hund", hvis formålet med det neurale netværk er at genkende objekter i billeder. Når billedet er kørt igennem det neurale netværk, giver netværket dets bud på hvilken kategori, billedet tilhører. Herefter udregnes forskellen mellem det beregnede og det ønskede resultat, hvilket kaldes krydsentropitabet. Det beregnede krydsentropitab fødes herefter baglæns ind i det neurale netværk og opdaterer vægtene i retning af det ønskede resultat.

I starten resulterer det neurale netværk ikke i andet end støj. Men ganske langsomt, iteration for iteration, lærer netværket at tilpasse sig det pågældende træningsdata. Når det beregnede resultat konvergerer mod det ønskede resultat, er træningen afsluttet og netværket er nu specialiseret i at klassificere datasættet. Hvis datasættet indeholder tilstrækkeligt mange annoterede billeder og er repræsentativt for de ønskede kategorier, for eksempel hunde og katte, har man nu en udmærket hunde- og kattedetektor.

Et netværk trænes ved at udregne dets respons for en række billeder


Kunstigt
neuralt
netværk
Krydsentropitab
→ Forward pass – klassifikation
← Backward pass – opdatering af vægte
○ Neuron

	Beregnet resultat	Ønsket resultat
Kat	0,323	0
Hund	0,202	1
Bil	0,475	0

Fig. K.3: Træningsprocess af et kunstigt neuralt netværk.

(grønne pile i figur K.3), hvor vi på forhånd har defineret det ønskede resultat (annoteret data). Forskellen mellem det ønskede resultat og det beregnede resultat beregnes i det såkaldte krydsentropitab, som føres baglæns gennem netværket for at opdatere dets vægte (røde pile i figur K.3).

To af nøgleordene bag denne læringsalgoritme er differentiabilitet og kædereglen for diffentiering af sammensatte funktioner. Alle de neuroner, som et kunstigt neuralt netværk er sammensat af, er grundlæggende (stykvist) differentierbare funktioner. Det betyder, at vi kan flytte det samlede netværks opførsel ved, neuron for neuron, at finde gradienten for den partielt differentierede funktion, opdatere funktionens vægte på baggrund heraf, og føre gradienten videre til de neuroner, som funktionen er forbundet til. Denne proces gentages for hver iteration, indtil alle neuroner er opdateret.

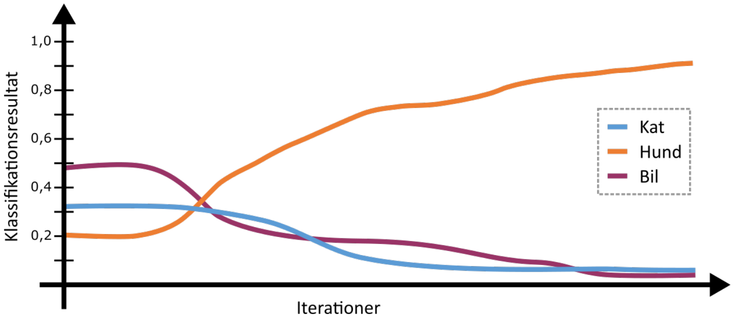


Fig. K.4: Iterativ læring af et kunstigt neuralt netværk. I starten resulterer netværket ikke i andet end støj, men jo flere annoterede billeder, der køres igennem netværket, jo bedre bliver det til at klassificere billeder af hunde som ”hund”.

4 HVORFOR FØRST NU?

Kunstige neurale netværk går tilbage til 1940'erne. Disse netværk var imidlertid ikke særlig populære i samtiden på grund af den beregningsmæssige kompleksitet og manglende træningsdata.

Den beregningsmæssige kompleksitet skyldes, at et netværk, der kan løse praktiske problemer, indeholder mindst tre lag med mange neuroner i hvert lag, der er internt forbundet med hinanden. Denne type netværk er kendt som en Multi-Layer Perceptron (MLP). I en MLP er hvert neuron i et lag forbundet til alle neuroner i det næste lag af netværket, som set i figur K.5. Dette fænomen, der er kendt som fuldt forbundne netværk, resulterer i store matricer, der beskriver vægtningen af neuronernes forbindelser. De store matricer fører igen til beregningsmæssigt krævende læringsalgoritmer, der i mange år var for store til, at computere kunne håndtere dem. Dette ændrede sig dog med introduktionen af såkaldte Graphics Processing Units (GPU) i 1990'erne, som tilbød hurtig og parallel databehandling. Brugen af GPU'er har gjort det muligt at implementere neurale netværk i praksis, hvorefter deres popularitet kun er steget. Faktisk er neurale netværk nu blandt de allerbedste værktøjer, der er i stand til at løse meget komplicerede problemer som billedbaseret objektgenkendelse. Denne succes skyldes imidlertid ikke kun udviklingen af bedre GPU'er, men også tilgængeligheden af enorme mængder data.

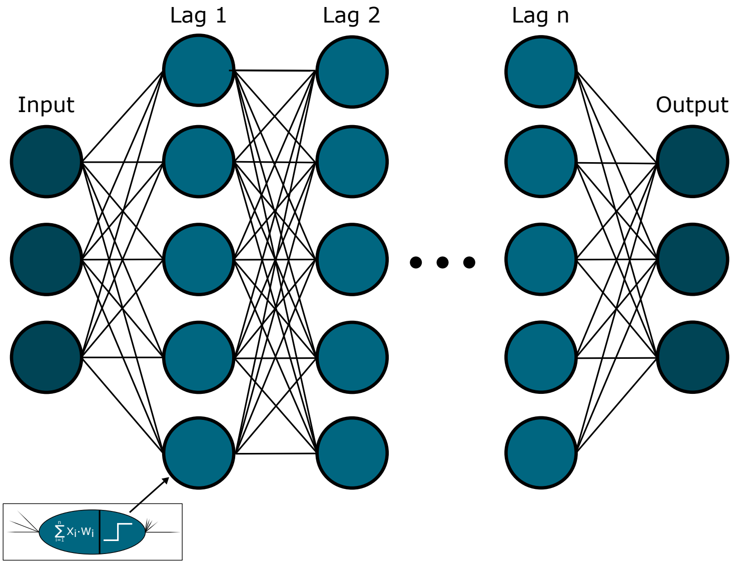


Fig. K.5: I et Multi-Layer Perceptron (MLP) netværk er alle neuroner forbundet til hinanden imellem lagene.

Da kunstige neurale netværk udelukkende kan lære på baggrund af eksempler, er tilgængeligheden af eksempeldata kritisk. Jo mere data, jo bedre er læringsprocessen. Imidlertid var store databaser ikke så almindelige for blot 10 år siden. Men siden 2010 er enorme databaser gradvist blevet opbygget. Et eksempel er ImageNet, der består af cirka 14 millioner annoterede billeder, inddelt i mere end 20.000 forskellige kategorier.

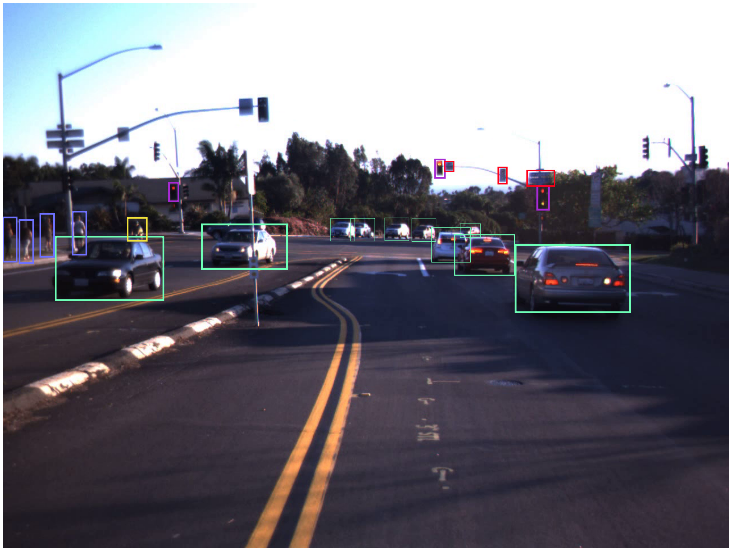


Fig. K.6: Overlejrede annoteringer af objekter i trafikken, hvor hver farve indikerer en kategori. Denne type data er fx vigtig for træningen af selvkørende biler.

De fleste billeder i ImageNet indeholder kun én annotering, det vil sige at hele billedet tilhører én kategori. En anden og mere omfattende måde at annotere billederne på er at definere det specifikke område i billedet, der indeholder et givent objekt – for eksempel fodgængere, cyklister, biler og trafikskilte, som ses i figur K.6.

5 FRA MACHINE LEARNING TIL DEEP LEARNING

Traditionelle machine learning-teknikker er baseret på at udvikle og udvælge specifikke karaktertræk, også kaldet features, ved de objekter, man ønsker at finde og genkende. Det har betydet, at forskere tidligere har brugt tid på manuelt at definere features, som efter deres vurdering var unikke og gav en god repræsentation af de ønskede objekter.

Til detektion af trafikskilte vil man i traditionel machine learning udvælge features, der kan beskrive skiltets cirkulære form og dets karakteristiske røde kant. Herefter udvælger man manuelt en eller flere metoder, der kan konvertere de ønskede features til en matematisk repræsentation. Disse features bruges til at træne en machine learning-algoritme, som benytter de udregnede features til at skelne mellem trafikskilte og ikke-trafikskilte.

Til trods for, at det i sidste ende er computerens algoritmer, der udregner det endelige resultat, indebærer traditionel machine learning en del manuelt arbejde med at definere hvilke features, der er relevante. Deep learning har til forskel fra machine learning ingen behov for menneskelig indblanding i forbindelse med udvælgelse og udformning af features. Sammenhængen mellem kunstig intelligens, machine learning og deep learning ses illustreret i figur K.7.

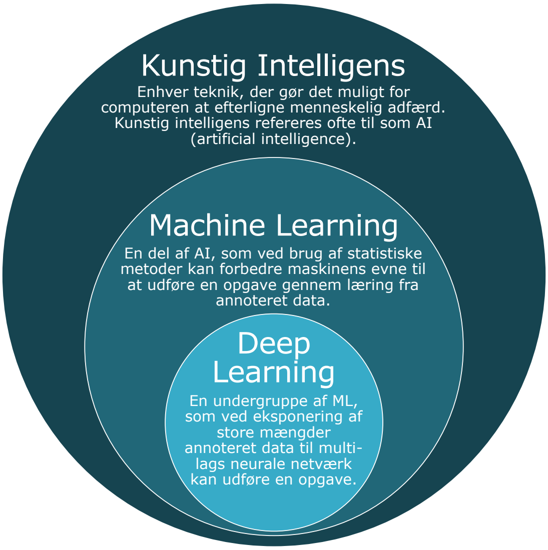

Kunstig Intelligens
Enhver teknik, der gør det muligt for computeren at efterligne menneskelig adfærd. Kunstig intelligens refereres ofte til som AI (artificial intelligence).
Machine Learning
En del af AI, som ved brug af statistiske metoder kan forbedre maskinens evne til at udføre en opgave gennem læring fra annoteret data.
Deep Learning
En undergruppe af ML, som ved eksponering af store mængder annoteret data til multi-lags neurale netværk kan udføre en opgave.

Fig. K.7: Sammenhæng mellem kunstig intelligens, machine learning og deep learning.

Det kræver dog stadig menneskelig indblanding, når deep learning-netværkene skal designes. Det gøres for eksempel ved at definere netværkets størrelse, det vil sige hvor mange lag, netværket skal bestå af. Workflowet fra mellem machine learning og deep learning er illustreret i figur K.8.

Et lag består af en række funktioner. Den vigtigste funktion i moderne neurale netværk er en såkaldt convolution (på dansk en foldning), og derfor kaldes disse netværk også Convolutional Neural Networks (CNN'er). Convolution er en matematisk operation, der benytter sig af et filter. Filtrenes overordnede funktion er at trække features ud af inputbilledet, og et moderne neuralt netværk indeholder rigtig mange filtre, der er grupperet i flere convolution-lag. Populære deep learning-netværk som AlexNet,

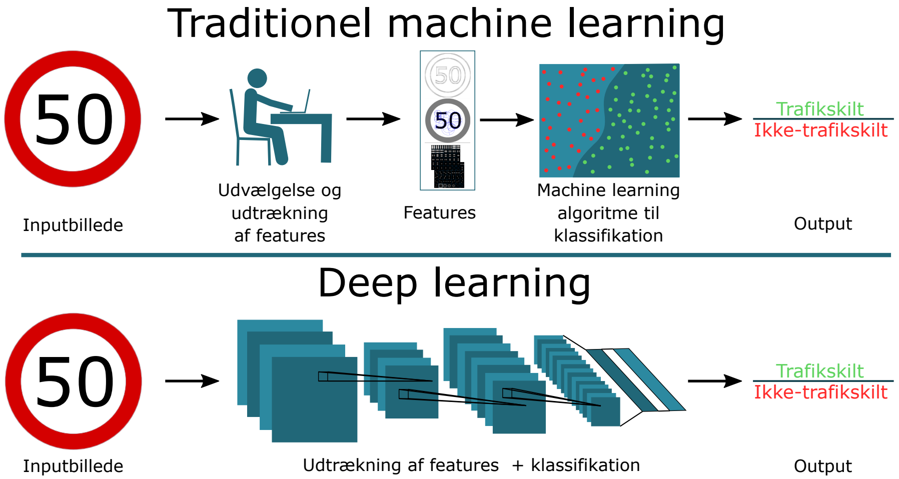

Traditionel machine learning
50
Inputbillede
Udvælgelse og udtrækning af features
Features
Machine learning algoritme til klassifikation
Trafikskilt
Ikke-trafikskilt
Output
Deep learning
50
Inputbillede
Udtrækning af features + klassifikation
Trafikskilt
Ikke-trafikskilt
Output

Fig. K.8: Sammenligning af workflowet for traditionel machine learning og deep learning. I modsætning til traditionel machine learning kræver deep learning ikke manuel udvælgelse af features.

VGG, GoogLeNet og Microsoft ResNet er alle CNN'er, og de indeholder henholdsvis 8, 19, 22 og 152 lag. Et eksempel på en convolution ses til venstre i figur K.9, hvor der benyttes et filter af størrelsen 3x3 pixels med udgangspunkt i den pixel, der er markeret med rød ring. Convolution består i, at man anvender 3x3 filteret på den "røde" centerpixel samt dets nabopixels, illustreret med det grå område i input- matricen. Den resulterende pixelværdi i outputmatricen opnås ved at gange filterets vægte på de tilhørende pladser i inputmatricen for derefter at summere resultatet. Herefter rykker vi vores 3x3 filter én gang til højre og gentager udregningen.

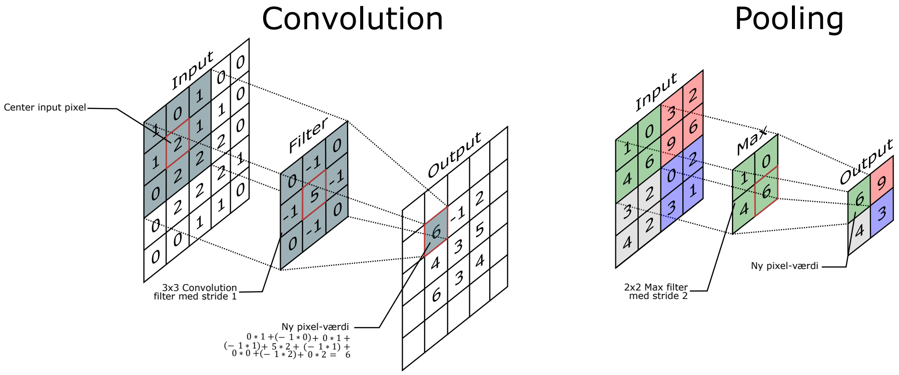


Fig. K.9: De vigtigste funktioner i moderne neurale netværk er convolution og pooling.

Outputtet fra convolution kaldes et feature map, og det er udgangspunktet for et andet meget anvendt lag i neurale netværk kaldet pooling, som ses til højre i figur K.9. I eksemplet på pooling bruges her et såkaldt max pooling-filter med en størrelse på 2x2 pixels og en stride på 2 pixels. En stride på 2 pixels betyder, at vi flytter filteret 2 pixels til højre efter hver operation. Max-filteret undersøger alle værdierne i et 2x2 område, tager den højeste værdi heri og smider de øvrige værdier væk. Den højeste værdi udgør nu den nye pixelværdi i outputmatricen. Denne funktion reducerer størrelsen på de feature maps, der genereres fra convolution-laget, således at man opnår en mere kompakt repræsentation.

En af årsagerne til, at CNN fungerer så godt, er, at netværkene selv kan lære at sammensætte både simple og komplekse features i deres convolution-lag – uden at man som operatør specifikt beder dem om at gøre sådan. Eksempel på både simple såvel som komplekse features ses i figur K.10.

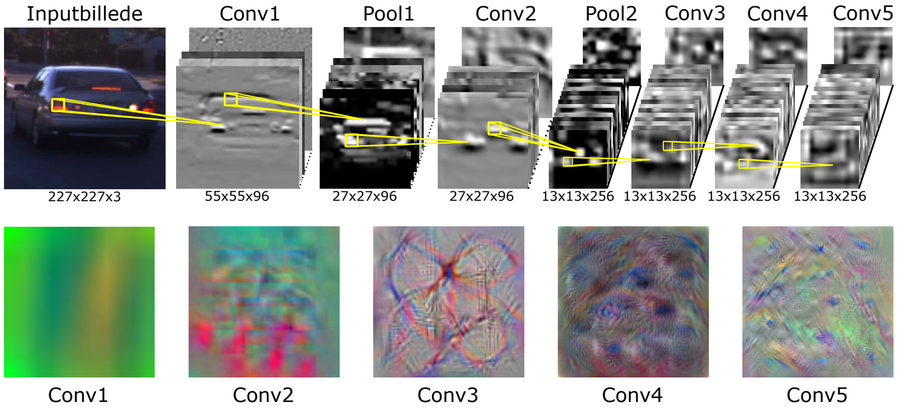


Fig. K.10: Eksempel på et AlexNet-inspireret neuralt netværk med 5 convolution-lag og 2 pooling-lag samt dertil hørende lærte features for hvert convolution-lag (øverst). Conv-lagene ændrer sig gradvis fra at være ret genkendelige omrids af bilen i conv1 til mere komplekse strukturer i conv5, som knap nok er genkendelige for det menneskelige øje. Den nederste del af figuren visualiserer, hvad 1 tilfældigt udvalgt feature-map i hvert af de 5 convolution-lag reagerer på i billedet. Conv1-features repræsenterer kanter og farveforskelle i forskellige retninger, mens de senere convolution-lag repræsenterer komplekse, specialiserede mønstre.

6 IKKE BEGRÆNSET AF MENNESKELIGE SANSER

Det ser i store træk ud til, at deep learning og kunstige neurale netværk virker som hjernen ved løsning af bestemte opgaver. Det betyder, at kunstig intelligens i princippet vil kunne klare det samme som et menneske. Potentialet er dog endnu større for den kunstige intelligens, da dets input ikke er begrænset til de menneskelige sanser, men vil kunne opfatte verden gennem et utal af sensorer og have adgang til ufattelige mængder information. For at nå så langt kræves der dog betydelige fremskridt indenfor udvikling af læringsalgoritmer og håndtering af information. På vejen dertil vil deep learning ændre fremtiden i mange forskellige applikationer og sektorer, fra sikkerhed og finans til medicin og transport. Vi er glade for at være en del af dette spændende eventyr.

Paper L

Techtunnel - Folkemøde 2018

Morten B. Jensen, Malte Pedersen & Poul Lund

This technical report has been used internally at Aalborg University,
September 2018

The layout has been revised.

1 FOLKEMØDET

Since 2011 Bornholm has hosted Folkemødet, directly translated to "The Peoples Meeting", which is a yearly recurrent event lasting four days, with political debates, happenings, and concerts. All the Danish political parties are invited to the event in Allinge and is given the opportunity to hold a major speech for the people. Beside the major speeches, there are a lot of open debates and discussions, which allow people to ask questions directly to politicians and experts.

As the admission is free, the meeting attracts a lot of different peoples, which makes it an obvious opportunity for companies, entrepreneurs and the likes to acquire information about what is stirring by engaging and participating in relevant debates. This is supported by the number of visitors, which reached a record high of 113,000 people visiting the meeting across the four days in 2018. A graph showing the estimated number of visitors every year since the beginning in 2011 can be seen in Figure L.1[1].

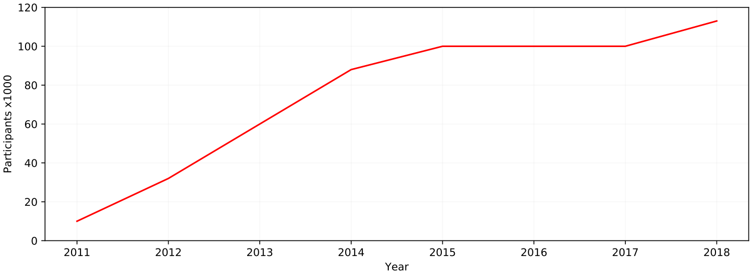


Fig. L.1: Graph showing the increase in participants over the years.

1.1 The Role of the University

The meeting offer the university a great opportunity to debate with people and politicians about the future technologies and how they may impact us. This includes the many aspects of computer vision, which is incorporated in more and more applications that people encounter every day. In most cases people without a technical background is not aware of where the technology is moving, what it can be used for and why it is important to pay attention.

The universities are obligated to inform people about the new technologies and Folkemødet is an obvious opportunity to do so. The meeting attracts people that comes from all walks of life, which opens for a variety of discussions about ethics, problematics and opportunities. Furthermore, the meeting

[1] https://folkemoedet.dk

has a high potential for promoting the engineering work conducted by the universities by showing off systems and applications in a digestible way. An example is shown in Figure L.2, where Malte Pedersen is explaining a group of listeners how deep neural networks can be used to detect objects in images, while a live demonstration is running on a monitor just outside of the photograph.

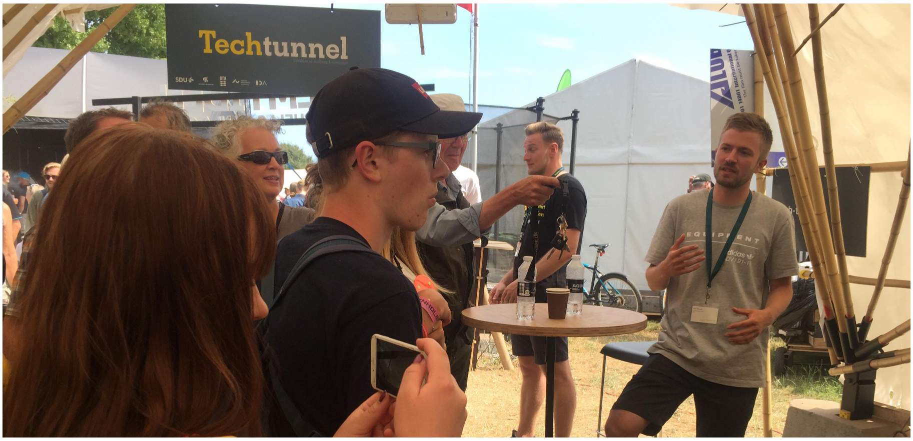


Fig. L.2: The stand of Aalborg University inside the Techtunnel, where Malte Pedersen describes how object detection works. Furthermore, a man is investigating whether the system is capable of detecting his key-hanger.

2 TECHTUNNEL

The meeting is split into several areas, that hosts different types of events and the main road of Allinge links them. However, as Allinge is a small city with only around 2,000 permanent residents, the regular infrastructure was estimated not to be sufficient to handle all the visitors. Therefore, the organizers of Folkemødet asked Aalborg University to create something that would draw people to use a shortcut between two of the major areas in order to reduce the traffic on the main road. The location of the roads and areas are illustrated in Figure L.3.

In order to attract people to drop by the Aalborg University stand, a large tunnel, known as the Techtunnel, was built on a pathway between two of the main areas. The tunnel was designed and created by Poul Lund using reusable bamboo-sticks held in place by custom made iron-racks and roofed with fire-resistant heat shrink plastic. The use of bamboo as a fundamental part was intended as the tunnel itself should represent a proposition of how the future includes both state-of-the-art technologies and traditional organic

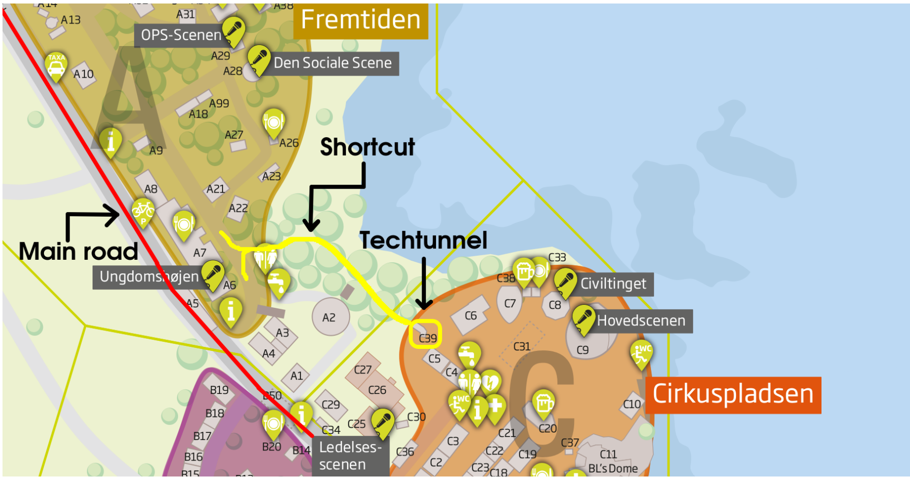


Fig. L.3: Overview of the two major areas, A and C, and the location of the shortcut and Techtunnel.

materials, such as the strong and eco-friendly bamboo. An image of the entrance to the tunnel can be seen in Figure L.4.

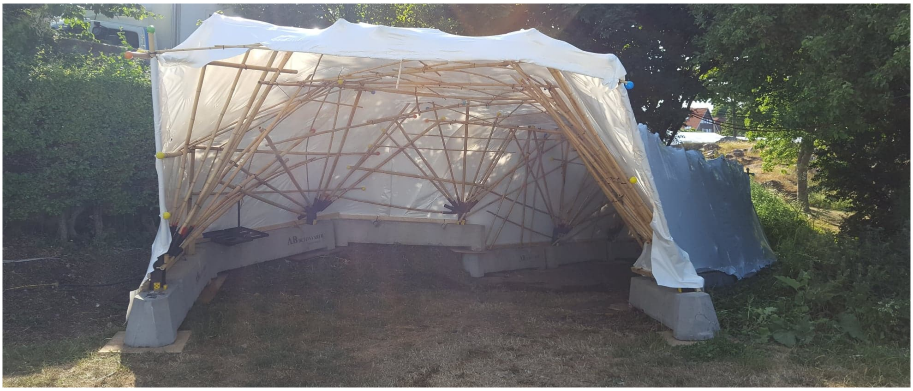


Fig. L.4: Entrance to the bamboo tunnel seen from area C.

Two monitors were placed in the tunnel as visual attractions and each of the them were connected to a computer and a webcam. The black rectangles connected to the camera-icons in Figure L.5 illustrate the placement of the monitors and cameras, respectively. When going from area C to area A, the visitor will see a monitor to the left of the entrance, which detects objects caught by the webcam. Moving further down the tunnel, another monitor appears which shows a single number representing the total amount of visitors that has passed through. Both systems will be described in further details in

Section 3.

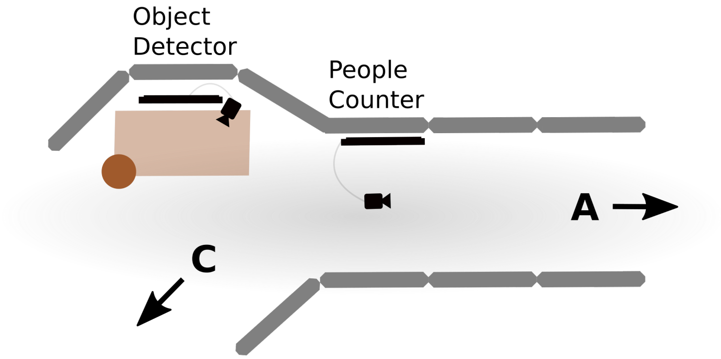
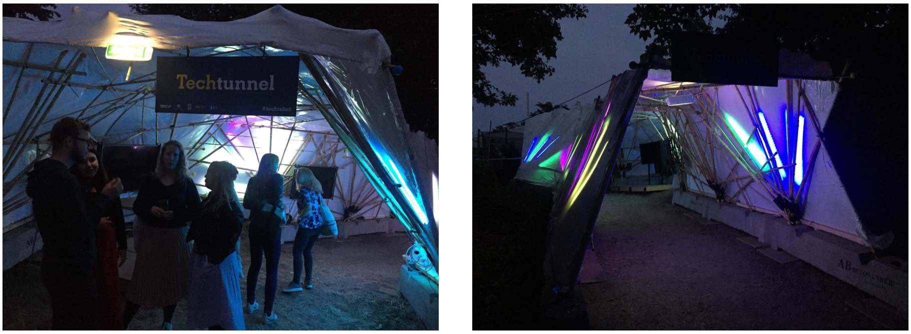


Fig. L.5: Setup of monitors and webcams.

Finally, the tunnel was also covered with RGB LED strips on the inside, which slowly faded through various colors to create a cozy atmosphere. The images presented in Figure L.6 shows the tunnel an evening seen from both ends. From Figure L.6a it can be seen that the tunnel was also a popular attraction in the evening, even though the monitors were turned off.



(a) Entrance from area C.

(b) Entrance from area A.

Fig. L.6: The Techtunnel seen from both ends in the evening.

3 COMPUTER VISION APPLICATIONS

The technical research conducted on the universities may be difficult to comprehend for people without technical knowledge. Therefore, a part of the project was to come up with ways to present complicated image processing software in a compelling and comprehensive way.

Two applications were made for the meeting, and the main prerequisite for both were some kind of interaction between the user and monitor. Both applications will be described briefly in the following sections.

3.1 Object Detector

The main attraction was an object detector based on the YOLOv3 deep neural network, which was used to analyze images in real-time from a webcam mounted above the monitor. When people walked into the tunnel and looked at the monitor, they would see them self being classified as a "Person" by the system. The network was trained to recognize 80 different items, so if the person had a backpack, a glass of beer, a bottle of water or another of the classes known to the network, they would be classified as well. An example can be seen in Figure L.7, where an image has been taken of the monitor while the system is running. Several people and objects are being recognized by the system illustrated by the labelled boxes surrounding the respective items.

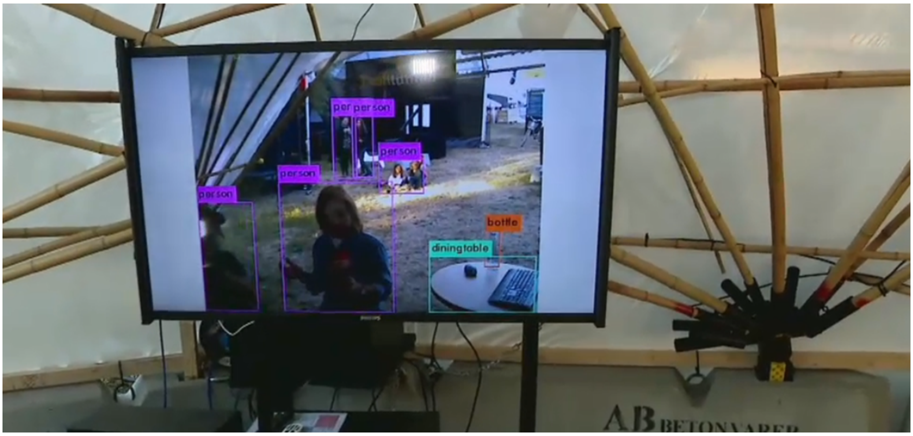


Fig. L.7: Picture of the system detecting a bottle, a dining table, and several persons.

The fact that people could see them self on the screen created an instant attraction for some and provocation for others and both were obvious opportunities to explain the listener how the system works and go into discussion about the pros and cons. As people recognize these types of systems from crime-series, movies and surveillance they are fast to join the conversation and may have positive or negative experiences and meanings they want to share and discuss.

3.2 People Counter

The second attraction was a people counter, which showed an estimate of the total number of visitors, that had passed through the tunnel. The webcam used for monitoring people was placed in the ceiling in the center of the tunnel, pointing towards area A as illustrated in Figure L.5. When people walked through the tunnel, the system would detect and track each individual, count them and increment the counter on the monitor accordingly.

The detection module was based on the YOLOv3 deep neural network and a Kalman filter was used to assist keeping track of the detections to avoid the same individual being counted more than once while being inside the field of view. An image of the monitor and people walking by it can be seen in Figure L.8.

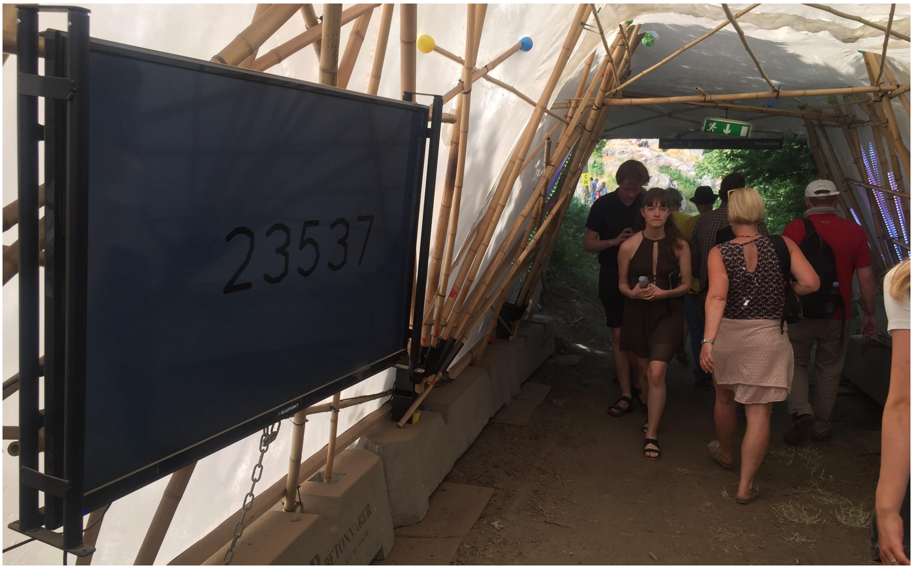


Fig. L.8: Picture of the people counter showing that a total of 23537 people had walked through the tunnel.

During the meeting, the people counter would log the amount of visitors every 5 minutes. A graph showing the increase of visitors over time is presented in Figure L.9. The graph has date and time on the horizontal axis and the total amount of visitors on the vertical axis. The blue and red regions are the on- and offline periods, respectively. It should be noted that both systems were shut down and locked up every night and no data was recorded on Sunday.

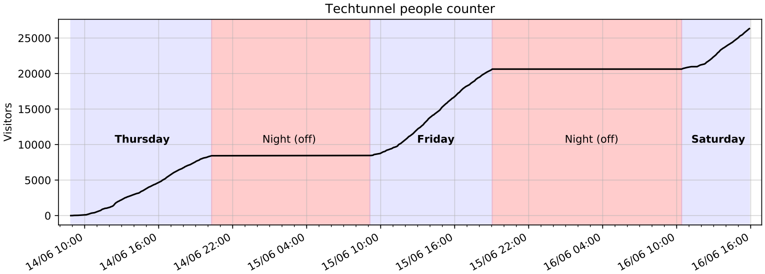


Fig. L.9: Graph showing the total amount of visitors over time. The blue and red regions show the on- and offline periods, respectively.

4 OUTCOME

Throughout Folkemødet the Techtunnel gathered attention from a lot of people. Even in the days before the meeting began, the local tv-station TV2 Bornholm visited the tunnel to make a piece about the risk of fire when mounting the shrink plastic and how the object detection worked[2]. As the meeting began the national news station TV2News dropped by to make a piece about Engineer the Future, which was a gathering of engineering sections from several institutions and universities. The journalist went through several of the other technologies but finished the piece with a several minutes long interview of Morten Bornø Jensen, who explained how the technologies behind the object detection and people counter worked. During the meeting several people mentioned that they had seen the tunnel in either TV2Bornholm or TV2News and dropped by to see it.

Throughout the entire meeting people visited the tunnel, used it as a shortcut or dropped by to get a status on how many people had went through the tunnel. Due to the tunnel being totally custom made in bamboo, steel, and concrete it differed from all the other tents and structures, which resulted in many comments and discussions about innovation, sustainability and creativity. Both seniors, kids, and adults asked questions about the technologies and they came from all social classes, so the debates and discussions varied and were in almost every case rewarding for both parts. All in all, an estimated 25,000 people went through the tunnel during the time the systems were running and there were never more than a few minutes without questions or discussions.

[2] http://play.tv2bornholm.dk/?area=specifikTV&id=747324

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-372-3

AALBORG UNIVERSITY PRESS