Vision-based Person Re-identification in a Queue

Lejbølle, Aske Rasch

Publication date:
2020

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

VISION-BASED PERSON RE-IDENTIFICATION IN A QUEUE

BY
ASKE RASCH LEJBØLLE

DISSERTATION SUBMITTED 2020

AALBORG UNIVERSITY
DENMARK

Vision-based Person Re-identification in a Queue

Ph.D. Dissertation

Aske Rasch Lejbølle

Dissertation submitted December XX, 2019

Dissertation submitted:	January 9, 2020
PhD supervisor:	Prof. Kamal Nasrollahi Aalborg University
PhD committee:	Associate Professor Claus B. Madsen (chairman) Aalborg University
	Senior Software Engineer Kristian Kirk CLAAS E-Systems
	Professor Ramalingam Chellappa UMIACS University of Maryland
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-582-6

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, 2020

Curriculum Vitae

Aske Rasch Lejbølle

In 2014, Aske Rasch Lejbølle received his BSc in Electronic Engineering and IT, followed by an MSc in Vision, Graphics and Interactive Systems in 2016. Both degrees were received from Aalborg University, Denmark. His master's project on Person Re-identification led to the commencement of the Ph.D. project in 2017 on the same topic, in collaboration with Veovo (formerly known as BLIP Systems A/S). During the Ph.D. project, Aske has spent three months at the University of California Riverside, where he was a part of the Video Computing Group led by Prof. Amit K. Roy-Chowdhury.

His research interest primarily involves computer vision and machine learning, more specifically, deep neural networks, which has been a key methodology of the Ph.D. project. During his time as a Ph.D. student, Aske has been involved in teachings, as well as, supervision of graduate and undergraduate students.

Curriculum Vitae

Abstract

The continuously growing aviation industry challenges airports around the world. For the airports to optimize the use of their staff, they need to know where and when staff is needed. One solution is to measure queue times in different process areas of the airport and use this knowledge to relocate staff.

In this Ph.D. thesis, we investigate vision-based person re-identification for queue time measurements. Using only two cameras, one at the queue entrance and another at the exit, we extract discriminative features from persons captured by both camera, and aim to find correspondences in order to measure queue times.

First, we present two novel overhead person re-identification datasets that were collected in queue scenarios using 3D cameras. The first dataset was collected at a university canteen, while the second was collected in real airport queue scenario.

Next, we propose a series of multimodal convolution neural networks that fuse RGB and depth information to a more robust multimodal feature representation. The networks are based on extracting both global and dynamically weighted local feature representations and fuse these for both RGB and depth before the two feature descriptors are fused to a multimodal one. The networks show state-of-the-art precision on three overhead re-id datasets. Additionally, through testing our proposed systems on the airport dataset, we show that median queue times based on re-identification deviates from the ground truth by only a small margin.

Thirdly, we consider re-identification from a more practical viewpoint, by proposing a method to transfer knowledge from an existing camera network to a newly introduced camera using already learned models and only few newly labeled samples in the expanded camera network. We show that the method outperforms related model learning methods that only use few labeled samples. Finally, we also consider, which edge platforms that can be used to deploy such a re-id system. Through evaluation of specific edge platforms on three different Computer Vision tasks, we show the potential of various platforms that can be purchased at reasonable prices.

Resumé

Den konstante udvikling inden for lufthavnsindustrien udfordrer lufhavne verden over. For at lufthavnene kan optimere brugen af deres personale, skal de vide hvor og hvornår de skal bruges. En metode til at løse dette, er at måle kø tider i forskellige process områder af lufhavnen, hvilket kan bruges til at flytte rundt på medarbejdere.

I denne Ph.D. afhandling udforsker vi vision-baseret person re-identifikation til måling af kø tider. Mere specifikt undersøger vi, hvordan man ved hjælp af kun to kameraer, et ved starten af køen og et andet ved slutningen, kan udtrække diskriminative karakteristika fra personer optaget af begge kameraer, og finde sammenfaldende karakteristika for derved at kunne måle kø tiden.

Først præsenterer vi to nye datasæt, der er optaget fra toppen i en kø kontekst ved hjælp af 3D kameraer. Det første datasæt er indhentet i en universitetskantine, og det andet i en lufthavn i et rigtigt kø scenarie.

Derefter foreslår vi en serie af multi modale neurale netværk (convolution nerual networks), der kombinerer RGB og dybde information til at skabe en mere robust representation. Netværkene er baseret på at udtrække både globalt og dynamisk vægtede lokalt information og kombinere disse for både RGB og dybde, inden de to modaliteter kombineres til en enkelt multi modal representation. Netværkene viser state-of-the-art præcision på tre re-id datasæt optaget fra toppen. Ved at teste vores system på lufthavnsdatasættet, viser vi derudover at median kø tider ved hjælp af re-identifikation kun afviger med en lille margin i forhold til de rigtige.

Vi betragter også re-identifikation fra et mere praktisk synspunkt ved at foreslå en metode, der kan bruges til at overføre viden fra et eksisterende kamera netværk til et nyligt introduceret kamera, ved brug af tidligere lærte modeller og kun enkelte kendte personer i det udvidede kamera netværk. Vi viser at metoden præsterer bedre end metoder, der kun benytter enkelte kendte personer. Til sidst undersøger vi også, hvilke platforme, der kan benyttes til at udrulle et re-id system. Gennem evaluering af specifikke edge platforme inden for tre forskellige Computer Vision opgaver, viser vi potentialet ved flere platforme, der kan købes til en overkommelig pris.

Contents

Thesis Details

Thesis Title:	Vision-based Person Re-identification in a Queue
Ph.D. Student:	Aske Rasch Lejbølle
Supervisors:	Prof. Kamal Nasrollahi, Aalborg University
	PhD Benjamin Krogh, Veovo Denmark

Part I and II of this thesis consist of an introductory overview and description of data collection, respectively. Meanwhile, Part III and IV of this thesis consist of the following papers that are either accepted or under review. Furthermore, an on-going work is included as a technical paper.

Feature Extraction

[A] Aske R. Lejbølle, Kamal Nasrollahi, Benjamin Krogh, and Thomas B. Moeslund, "Multimodal Neural Networks for Overhead Person Re-identification," *Proceedings of the 2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 25–34, 2017.

[B] Aske R. Lejbølle, Benjamin Krogh, Kamal Nasrollahi, and Thomas B. Moeslund, "Attention in Multimodal Neural Networks for Person Re-identification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 179–187, 2018.

[C] Aske R. Lejbølle, Kamal Nasrollahi, Benjamin Krogh, and Thomas B. Moeslund, "Person Re-identification Using Spatial and Layer-Wise Attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[D] Aske R. Lejbølle, Kamal Nasrollahi, and Thomas B. Moeslund, "Enhancing Person Re-identification by Late Fusion of Low-, Mid-, and High-Level Features," *IET Biometrics*, vol. 7, no. 2, pp. 125–135, 2018.

Practical Re-identification

[E] Sk Miraj Ahmed, Aske R. Lejbølle, Rameswar Panda, and Amit K. Roy-Chowdhury, "Camera On-boarding for Person Re-identification using Hypothesis Transfer Learning," Submitted to the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[F] Aske R. Lejbølle, Benjamin Krogh, Kamal Nasrollahi, and Thomas B. Moeslund, "One-to-One Person Re-identification for Queue Time Estimation," *Technical Report*, this paper is on-going work, 2019.

[G] Aske R. Lejbølle, Christoffer Bøgelund Rasmussen, Kamal Nasrollahi, and Thomas B. Moeslund, "Evaluation of Edge Platforms for Deep Learning in Computer Vision," Submitted to the journal of Neural Computing and Applications, 2019.

Preface

This Ph.D. study is a collaboration between the Visual Analysis of People (VAP) laboratory at the Section of Media Technology, Aalborg University, Denmark and Veovo Denmark, who excels in passenger predictability through, among other things, solutions to optimize queue management. The thesis resolves around person re-identification to measure queue times, by examining three areas; data acquisition, feature extraction, and practical re-identification. Part one introduces the topic and provides an overview of work conducted during the Ph.D. study. Part two describes the datasets that have been collected as part of the study, followed by two parts containing papers that deal with feature extraction and practical re-identification, respectively. Thus, this thesis is submitted as a collection of papers in partial fulfillment of a Ph.D. study at the Section of Media Technology, Aalborg University, Denmark and contains both published and currently reviewed work.

The work is carried out in the period Jan. 2017-Dec. 2019, in part at the VAP laboratory and in part at Veovo Denmark. The project, further, included a research stay in the Video Computing Group laboratory at the University of California, Riverside.

I would like to thank Prof. Kamal Nasrollahi, who has been my supervisor since the beginning of my adventure on person re-identification on the second semester of my master's, and has always brought inspiring ideas, while also allowing me to pursue my own ideas. Also thanks to Prof. Thomas B. Moeslund for taking over as supervisor during the second year, and bringing in new thoughts on the project. Thanks also to Prof. Amit K. Roy-Chowdhury, for allowing me to visit his laboratory and take part in excellent work.

I would also like to thank the team at Veovo Denmark, most of all my supervisor Benjamin Krogh for his great interest and guidance throughout the project, but also Mike Røntved, who has assisted me to get some of my work done faster. A big thanks also goes to my colleagues at VAP for their interest and great discussions on topics related to my project.

Aske R. Lejbølle
Aalborg University, January 9, 2020

Part I

Overview

Chapter 1

Introduction

The aviation industry has been growing vastly within the last couple of decades. Passenger journeys have increased from an estimated 1.67 Billion in 2000 to an estimated 4.23 Billion in 2018 [1], a number which is expected to almost double by 2036 [2]. This growth will be mostly driven by China, the US, India, Indonesia and Turkey, as shown in Figure 1.1.

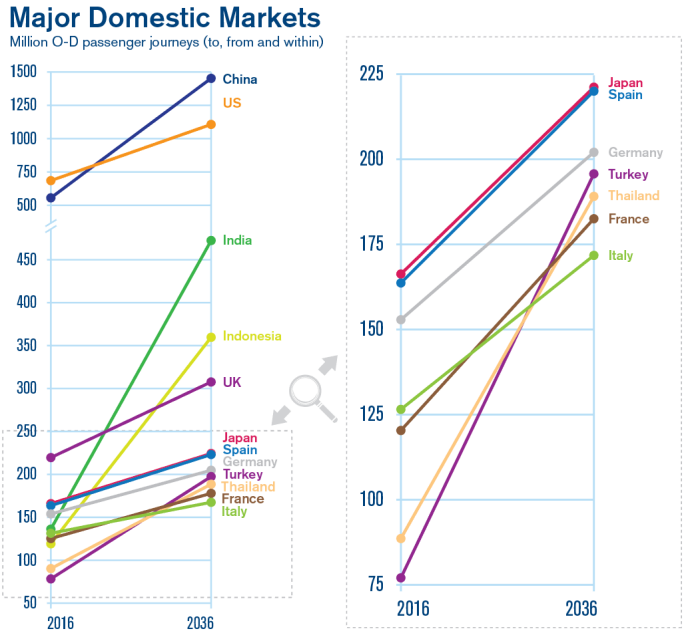


Fig. 1.1: Passenger growth forecast 2018-2036. *Image from [2] © 2017 IATA.*

The growing demand for transportation by air, challenges airports throughout the world, and results in needs to expand the airports to increase passenger capacity. Furthermore, airports are required to improve efficiency of the different processes, such as baggage handling, check-in and security checks, to maintain and improve passenger experience. From a Global Passenger Survey (GPS) in 2018 [3], passengers mentioned automation of airport processes and tracking on bags as desired capabilities. More importantly, a queue time of less than 10 minutes in security and immigration is desired. Standing in a queue is an everyday thing to people all around the world, which makes this demand relatable. A smaller queue time not only results in happier passengers, but also allows the airport to increase revenue as they can more easily manage and relocate staff as needed. Reducing queue times not only applies to airports, but amusement parks and ski resorts, which also in recent years have experienced increasing numbers of visitors [4, 5].

Some airports already deploy technology to track passengers and measure queue times, to optimize airport staff allocation and reduce queue times. Based on requirements, a number of technologies exist; (1) technologies that are based on WiFi/Bluetooth (BT) device tracking sensors to track passengers throughout the queue [6], (2) technologies that use existing airport closed-circuit television (CCTV) cameras to count the number of passengers within queues [7], (3) technologies that use newly deployed overhead cameras to track passengers throughout the queue [8, 9], and (4) technologies that fuse the use of overhead cameras and WiFi/BT to count and track passengers [6]. While these technologies might have the required capabilities, they do have shortcomings. A WiFi/BT device tracking solution is cheaper than camera-based solutions, however, since the introduction of randomized MAC addresses in iOS8 [10] and Android Marshmallow (v. 6.0) [11] in 2014 and 2015, respectively, the challenge of tracking passengers using this technology has increased. Using existing CCTV cameras is likely the cheapest camera-based solution. This requires the queue to be within the field of view of the cameras, thus, constraints the position of the camera. A more costly solution will then be to set up additional cameras, which is the case for most camera-based solutions. In high-ceiling areas, this is an optimal solution since only few cameras are required to cover a large area. In low-ceiling areas, on the other hand, this is costly due to the need for a large number of cameras, depending on the field of view of the cameras. A solution to this could be to use a combination of a few cameras to count the number of passengers and WiFi/BT sensors to provide device tracking data.

In this thesis, an alternative novel way to measure queue times, which uses only vision-based methods, is proposed. More specifically, a person re-identification (re-id) based approach is proposed, which measures the queue time of a passenger using images captured by only two non-overlapping cameras, one at the queue entrance and one at the queue exit, as shown in Fig-

ure 1.2. Characteristics, i.e., features, are then extracted from captured images and stored along with timestamps and ids. The goal is then to find matching characteristics from the two cameras, and use the corresponding timestamps to measure the queue times. Despite the large number of applications for which vision-based re-id can used to measure queue times, this thesis solely focus on re-id to measure queue times in an airport.

In the following, a general re-id pipeline is presented, followed by the main hypothesis of this thesis and an overview of the work that has been conducted to accept or reject this hypothesis. In the remaining of the thesis, *persons* and *passengers* are used interchangeably.

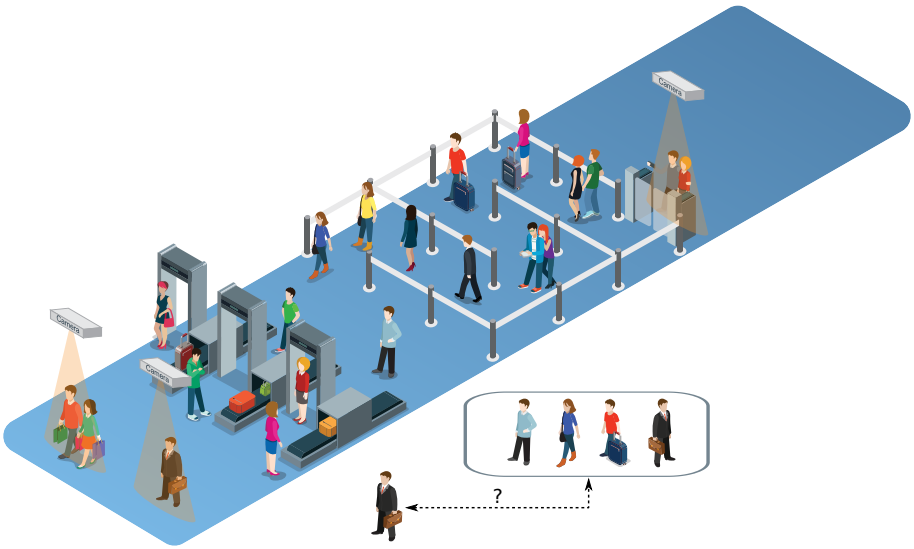


Fig. 1.2: Principle of the vision-based re-identification of passengers in a queue, which can be used to measure queue times. © Veovo

1 The Re-identification Pipeline

Figure 1.3 shows a general person re-id pipeline, which consists of several tasks. In general, person re-id is defined as matching features that are extracted from images of persons across non-overlapping cameras. In the following, each task will be briefly introduced.

Data Acquisition The first part of the pipeline, naturally, is acquisition of data, or more specifically, images. Data acquisition is performed by a sensor, i.e., camera, which captures images that are propagated through the rest of the pipeline. In case of airports, and surveillance in general, CCTV cameras

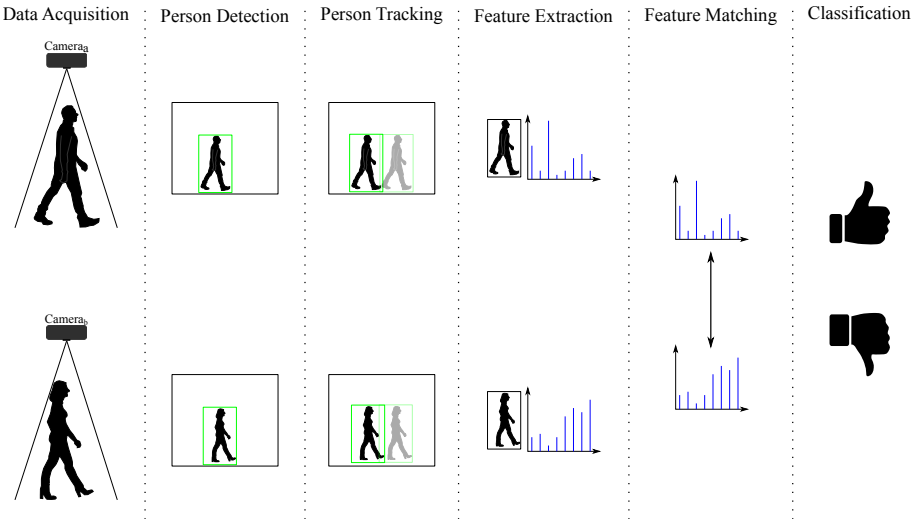


Fig. 1.3: A general person re-identification pipeline. Cameras *a* and *b* follow the same initial pipeline and are joint at the *Feature Matching* process.

are used to monitor check-in halls or security areas, and output a stream of images. In case of sensors to count passengers crossing certain areas of the airport, more intelligent cameras are deployed that can output coordinates of moving passengers [12, 13]. Depending on the purpose of the camera, it is also possible to deploy cameras that capture and output depth information [14, 15] or cameras that capture and output heat signatures [16].

Person Detection Correctly detecting passengers in images is crucial to extract robust feature descriptors that do not contain any noise. Here, object detection algorithms are used to detect a region of interest (ROI), within the image where an object, in this case a passenger, appears. Before the beginning of the era of deep learning in 2012, some of the most common object detectors were Histogram of Oriented Gradients (HOG) [17] and Deformable Part Models (DPM) [18]. Since 2012, state-of-the-art object detectors have been based on deep learning and divided into two subcategories; (1) one-stage detectors that detect the objects in a single-stage end-to-end fashion [19–21], and (2) two-stage detectors that first proposes regions to perform detection, followed by the detection itself [22–25].

Person Tracking To have a more robust feature descriptor, it is essential to capture several images of each passenger from which features can be extracted. In order to not manually having to go through all images to find detections of the same passenger, tracking algorithms are deployed to follow

the ROI around a passenger over the course of several frames. Tracking can be divided into two groups; (1) tracking by detection, where an object detector is applied to every frame and tracking trajectories are formed if an object is detected in multiple frames [26–28], and (2) detection-free tracking, which models the appearance of an object based on an initial detection and tries to track that object in subsequent frames [29–31].

Feature Extraction The most important task of the pipeline is the feature extraction, as having discriminative features is essential to correctly re-id passengers. Like detection and tracking, we can split feature extraction into two subcategories; (1) hand-crafted features, where features are devised based on internal structures of the image, such as colors, edges or corners, and (2) deep features, which have become increasingly popular since 2012 [32]. For deep features, a Convolution Neural Network (CNN) is implemented, which is trained end-to-end and combines feature learning and classification by forward propagating images of persons and output a label prediction. The predictions are then compared to the ground truth labels and the parameters of the network are updated based on the correctness of the predictions. Thus, feature learning is seen more as a black box.

Feature Matching Two types of feature matching are, typically, used in re-id; (1) Euclidean distance, which basically measures the sum of distance between points of feature vectors, and (2) Mahalnobis, which is based on the variance of feature dimensions and adds a covariance matrix, M, to distance calculations. In both cases, features from persons captured by a camera a are matched against features of persons captured by a camera b. Methods exist that enhance the performance of both metrics. For Euclidean distance, works have been presented that map features from the two views to shared feature spaces by learning a projection matrix [33–36]. The projection matrix is learned such that features of similar persons appear closer and those of dissimilar appear further away. Meanwhile, works have also been proposed, where the covariance matrix in Mahalanobis distance is learned based on similar and dissimilar feature pairs along with binary labels indicating relations [37–41]. The formerly mentioned work can also be categorized by a single category, *distance metric learning*.

Classification Person re-id is approached as either an image retrieval problem or a verification problem. In both cases, features extracted from a person in camera b, i.e., a probe, is matched against features from all persons seen in camera a, i.e., a gallery. In case of an image retrieval problem, classification outputs a list of likely matches, ranked by similarity. That is, the most likely match is ranked 1, the second most likely is ranked 2, etc. This case, hence,

assumes that a true match is somewhere in the ranked list, this is also referred to as a *closed-world* setting. Meanwhile, in the verification problem, features of a person in camera a is matched against features of persons in camera b, and for each matching, a binary output is provided indicating whether features represent the same (1) or different (0) persons. This technique is more suitable to match persons in cases where not all persons in camera a were necessarily in camera b. This is also referred to as an *open-world* setting.

2 Scope of this Thesis

The main scope of this thesis is to uncover whether vision-based re-id can be used to measure queue times, thus, we wish to accept of reject the hypothesis that *vision-based person re-identification can be used to correctly measure queue times.*

Ideally, this involves setting up the entire pipeline, as shown Figure 1.3, however, since each task in the pipeline is in itself a major research area, this thesis focus on the most important parts of the pipeline. More specifically, the focus of this thesis is to devise features that are as discriminative as possible and invariant to environmental changes, such as illumination variations. Furthermore, due to the novelty of the problem of using re-id to measure queue times, data acquisition is important to properly identify and evaluate feature robustness. Finally, this thesis also focus on re-id from a practical viewpoint to conclude how a re-id system to measure queue times can be deployed.

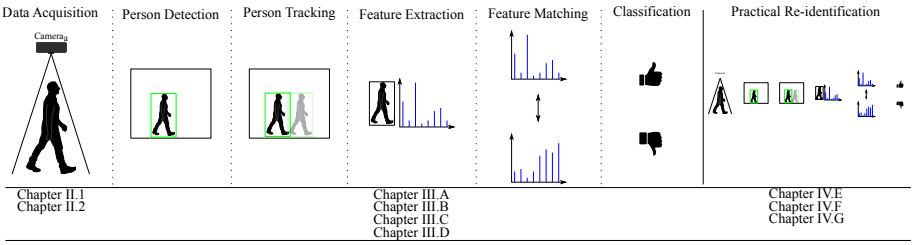


Fig. 1.4: The re-id pipeline related to the work presented in this thesis with focus on Data Acquisition, Feature Extraction and Practical Re-identification.

Figure 1.4 shows an overview of the articles and chapters that cover the work of this thesis in relation to the general re-id pipeline. Part II covers the data acquisition, which includes descriptions of two datasets that have been collected as part of the Ph.D. study; one at a university canteen (II.1) and another at an airport (II.2). Part III covers the feature extraction, including work where novel features are devised based on the re-id setting (III.A, III.B and III.C), and work that shows the complementarity of fusing the classification results of different features (III.D). Finally, apart from the pipeline,

Part IV covers re-id from a more practical viewpoint. This practical viewpoint includes work that proposes a method to transfer deployed models to new environments (IV.E), work that proposes an optimization of re-id for queue measurements (IV.F), and work that analyzes specific hardware platforms that can be used to run re-id on the edge (IV.G).

In the following chapters, each part is introduced, including a description of related state-of-the-art methods and a highlighting of key contributions.

References

[1] C. A. S. o. t. W. International Civil Aviation Organization and I. staff estimates. (2018) Air transport, passengers carried. https://data.worldbank.org/indicator/IS.AIR.PSGR. The World Bank Group. Accessed: November 27, 2019.

[2] I. C. Communications. (2017, October) 2036 forecast reveals air passengers will nearly double to 7.8 billion. https://www.iata.org/pressroom/pr/Pages/2017-10-24-01.aspx. IATA. Accessed: November 27, 2019.

[3] ——. (2018, October) Passengers want more information, automation, control & privacy but human touch still important. https://www.iata.org/pressroom/pr/Pages/2018-10-02-02.aspx. IATA. Accessed: November 27, 2019.

[4] M. Soberman. (2019, May) Tea and aecom release 2018 theme park attendance statistics, magic kingdom is worlds most visited park. https://wdwnt.com/2019/05/tea-and-aecom-release-2018-theme-index-and-museum-index-magic-kingdom-is-worlds-most-visited-park/. WDW News Today. Accessed: November 27, 2019.

[5] L. Vanat. (2019, April) 2019 international report on snow & mountain tourism. https://vanat.ch/RM-world-report-2019.pdf. Accessed: November 27, 2019.

[6] Veovo. (2019) Unleash the power of predictive insights. https://veovo.com/platform/passenger-predictability/. Veovo. Accessed: November 27, 2019.

[7] Foxstream. (2019) People counting and waiting time measurement. https://www.foxstream.us.com/flow-management/. Foxstream. Accessed: November 27, 2019.

[8] CrowdVision. (2019) Solutions for airports. https://www.crowdvision.com/solutions-airports/. CrowdVision. Accessed: November 27, 2019.

[9] Xovis. (2018) Security checkpoint. https://www.xovis.com/solutions/detail/security-checkpoint/. Xovis. Accessed: November 27, 2019.

[10] J. Cox. (2014, June) ios 8 mac randomizing just one part of apples new privacy push. https://www.networkworld.com/article/2361846/ios-8-mac-randomizing-just-one-part-of-apple-s-new-privacy-push.html. Network World. Accessed: November 27, 2019.

[11] A. Developers. (2015) Android 6.0 changes. https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html#behavior-hardware-id. Google. Accessed: November 27, 2019.

[12] Intenta. (2018) Intenta s2000. https://www.intenta.de/en/sensor-systems/intenta-s-2000.html. Intenta. Accessed: November 27, 2019.

[13] FLIR. (2019) Brickstream 3d gen 2. http://www.brickstream.com/Products/home-3DGen2.html. FLIR. Accessed: November 27, 2019.

[14] I. RealSense. (2018) Intel realsense depth camera d435. https://www.intelrealsense.com/depth-camera-d435/. Intel. Accessed: November 27, 2019.

[15] M. Azure. (2019) Azure kinect dk. https://azure.microsoft.com/en-in/services/kinect-dk/. Microsoft. Accessed: November 27, 2019.

[16] Hikvision. (2019) Ds-2td2166-7/15/25/35/v1 thermal network outdoor bullet camera. https://us.hikvision.com/en/products/cameras/thermal-camera/outdoor-bullet/high-resolution/thermal-network-outdoor-bullet-camera. Hikvision. Accessed: November 27, 2019.

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

[18] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*. IEEE, 2008, pp. 1–8.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. CVPR*, 2017, pp. 7263–7271.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. ICCV*, 2017, pp. 2980–2988.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014, pp. 580–587.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, 2017, pp. 2117–2125.

[26] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Proc. ECCV*. Springer, 2016, pp. 100–111.

[27] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese cnn for robust target association," in *Proc. CVPR*, 2016, pp. 33–40.

[28] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. AVSS*. IEEE, 2017, pp. 1–6.

[29] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, 1981, pp. 121–130.

[30] M. Danelljan, G. H´ager, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. BMVC*, 2014.

[31] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. CVPR*, 2019, pp. 1328–1338.

[32] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[33] S. Liao and S. Z. Li, "Efficient psd constrained asymmetric metric learning for person re-identification," in *Proc. ICCV*, 2015, pp. 3685–3693.

[34] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. CVPR*, 2016, pp. 1239–1248.

[35] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *Proc. ICCV*, 2017, pp. 994–1002.

[36] J. Dai, Y. Zhang, H. Lu, and H. Wang, "Cross-view semantic projection learning for person re-identification," *Pattern Recognition*, vol. 75, pp. 63–76, 2018.

[37] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. CVPR*. IEEE, 2009, pp. 498–505.

[38] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[39] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Person Re-Identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 12, pp. 247–267.

[40] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[41] Y. Yang, S. Liao, Z. Lei, and S. Z. Li, "Large scale similarity learning using similar pairs for person verification," in *Proc. AAAI*, 2016, pp. 3655–3661.

Chapter 2

Data Acquisition

As presented in section I.1.2, the first part of the thesis introduces the acquisition of data to devise and evaluate features.

1 Motivation

The majority of work in person re-id consider data captured from a primarily horizontal viewpoint as shown in Figure 2.1, however, in case of re-id in a queue this is impractical due to various reasons. Horizontally placed cameras are less discrete and if spotted by passengers, the passengers might feel monitored. More importantly, as shown in Figure 2.1 (a) the probability of occlusion is much higher with a horizontal viewpoint, which especially applies to passengers in a queue, if they follow a maze. Furthermore, re-id is often considered in the context of forensics. In that case, persons often move freely around in big environments monitored that by the cameras, while passengers in a queue are often constrained to follow a certain maze directed either by queuing barriers or by an airport staff.

As a result of aforementioned differences, this thesis focus on data captured from an overhead viewpoint. In contrast to a horizontal viewpoint, data captured from an overhead viewpoint potentially results in self-occlusion, which leads to less features of passengers being visible. To counter this, additional complementary data from other modalities are considered. Within computer vision (CV), typical options are either depth data collected from a stereo camera, or thermal data collected from a thermal camera. Since the cameras are placed overhead pointing downwards, the obvious solution is to capture additional depth information. The overhead depth information also allows us to capture the height of passengers, which is a potentially useful feature to combine with color and texture features.

Fig. 2.1: Examples images from (a) CUHK01 [1], (b) Market1501 [2], (c) MSMT17 [3] and (d) RAiD [4], captured from a primarily horizontal viewpoint. *(b), (c) and (d) are also used in [5].*

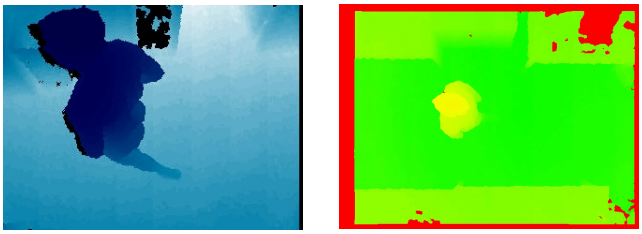
2 Related Work

As mentioned in section I.2.1, the majority of existing re-id datasets were collected from a primarily horizontal viewpoint. The early ones, typically, contain images that were collected across two non-overlapping cameras [6–8], more recently, large datasets were collected that contain images of more than thousand different persons collected from up to 15 cameras [2, 3, 9]. Detailed descriptions of person re-id datasets, primarily collected from a horizontal viewpoint, can be found in [10].

Only few datasets have been published that were collected using an overhead camera [11, 12], both using a single camera. [12] collected a Depth-based Person Identification from Top (DPI-T) dataset in a hallway using a single RGB-D camera[1]. The dataset contains images of 12 different persons, with each person appearing in up to five different sets of unique clothing. As the camera was placed indoor, the dataset contains less illumination variations. Nonetheless, movements of the persons were unconstrained and persons were also recorded whilst holding objects, such as plates or a cup of coffee. Another RGB-D based dataset captured from an overhead view was presented in [11] named Top View Person Re-identification (TVPR). The data were collected in a university office using an Asus Xtion Pro Live RGB-D camera [13], which was placed at a height of 4 m above the floor. The camera captures color images in SXVGA resolution (1280×1024), while it uses an infrared sensor to measure depth, which results in a depth map of size 640×480. 100 different persons were recorded over the course of eight days, causing the presence of illumination variations. While the movements of persons in [12] were unconstrained, the persons in [11] were instructed to follow

[1]The authors have not specified which camera was used.

a path directly below the camera, from left to right and vice versa. Examples of depth images from the two previously published datasets are shown in Figure 2.2. The datasets are used to evaluate features in [14–16].



(a) (b)

Fig. 2.2: Examples of depth images from (a) DPI-T [12] and (b) TVPR [11]. *Image from [14].*

Other RGB-D based datasets for re-id are publicly available [17–19], however, similar to most RGB-based datasets, they were collected from a horizontal viewpoint.

3 Contributions

Over the course of the Ph.D. project, two datasets were collected, both using RGB-D cameras from an overhead viewpoint. The first dataset was collected at a university canteen using a single RGB-D camera in an uncontrolled environment, and the second was collected at an airport using two non-overlapping RGB-D cameras. As defined in [14], the first dataset will henceforth be referred to as Overhead Person Re-identification (OPR) dataset, while the second, as defined in [20], will be referred to as Queue Person Re-identification (QPR).

While both datasets were collected in the context of queues, the first considers entrance and exit points to be the same, while the second is more realistic in terms of having entrance and exit at two separate locations with varying lighting and height. For both datasets, we used ZED cameras from stereolabs [21] that are able to capture images in up to 2k resolution (2048×1080 pixels). Since the ZED camera is a passive stereo camera, the resolution of the depth map is dependent on that of the captured RGB images, more details will be given in Part II.

Compared to previous overhead datasets, the datasets collected in this project are of much higher quality due to a higher resolution, which results in much more detailed information of persons, in terms of both color, texture and depth. Furthermore, in contrast to TVPR, OPR was collected in a much more uncontrolled environment with more diverse movement of persons, while QPR was collected from two non-overlapping cameras with large

variations in illumination. Compared to DPI-T, OPR and QPR contain higher numbers of persons, while the context is also more similar to that of re-id in a queue. For both OPR and QPR, we recorded timestamps that can be used to compare measured queue times using re-id with actual ones.

OPR is used to evaluate features in [14–16] (chapter III.B-III.D), while QPR is used in [20] to evaluate features and perform queue time measurements using re-id. Due to government legislation, it has not been possible to publish the datasets.

To summarize, the contributions of *Data Acquisition* includes:

- We collected two overhead RGB-D based datasets using high-resolution cameras to capture fine-grained details of both color, texture and depth. Both datasets were collected in uncontrolled environments.
- Through experiments in [14–16] we show that OPR is a more complex and difficult dataset to solve compared to previously published TVPR and DPI-T datasets.
- In [20] we use QPR to evaluate queue time measurements using vision-based re-identification.

References

[1] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[2] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. ICCV*, 2015, pp. 1116–1124.

[3] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person trasfer gan to bridge domain gap for person re-identification," in *Proc. CVPR*, 2018, pp. 79–88.

[4] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network," in *Proc. ECCV*. Springer, 2014, pp. 330–345.

[5] S. M. Ahmed, A. R. Lejbølle, R. Panda, and A. K. Roy-Chowdhury, "Camera on-boarding for person re-identification using hypothesis transfer learning," November 2019, under review for the 2020 IEEE Conference on Computer Vision and Pattern Recognition.

[6] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[7] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification." in *Proc. BMVC*, 2011, pp. 1–6.

[8] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning." in *Proc. ACCV*. Springer, 2012, pp. 31–44.

[9] M. Gou, S. Karanam, W. Liu, O. I. Camps, and R. J. Radke, "Dukemtmc4reid: A large-scale multi-camera person re-identification dataset." in *Proc. CVPR Workshops*, 2017, pp. 1425–1434.

[10] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[11] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, "Person re-identification dataset with rgb-d camera in a top-view configuration," in *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 1–11.

[12] A. Haque, A. Alahi, and L. Fei-Fei, "Recurrent attention models for depth-based person identification," in *Proc. CVPR*, 2016, pp. 1229–1238.

[13] Asus. (2012) Xtion pro live. https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/overview/. Asus. Accessed: November 28, 2019.

[14] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[15] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "Attention in multimodal neural networks for person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 179–187.

[16] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[17] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino, "Re-identification with rgb-d sensors," in *Proc. ECCV*. Springer, 2012, pp. 433–442.

[18] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. Van Gool, "One-shot person re-identification with a consumer depth camera," in *Person Re-Identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 8, pp. 161–181.

[19] M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti, "3d reconstruction of freely moving persons for re-identification with a depth sensor," in *Proc. ICRA*. IEEE, 2014, pp. 4512–4519.

[20] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "One-to-one person re-identification in a queue," Aalborg University, Tech. Rep., 2019.

[21] Stereolabs. (2017) Zed - depth sensing and camera tracking. https://www.stereolabs.com/zed/. Stereolabs. Accessed: November 28, 2019.

Chapter 3

Feature Extraction

Following Figure 1.4, the second part of the thesis deals with feature extraction, specifically, devising of discriminative features based on the re-id setting and the acquired data.

1 Motivation

As mentioned in section I.2.1, most prior re-id datasets were collected from a horizontal viewpoint and features have, thus, been devised based on frontal images of persons. Since we consider an overhead viewpoint, features devised for the horizontal view not necessarily work as well. As a result, it has been necessary to devise features that are robust when the person is seen from above and might inflict self-occlusion.

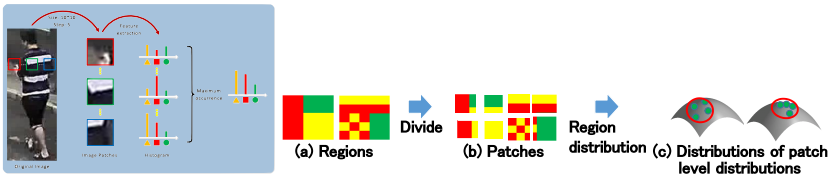
When devising novel features, it is also important to consider the type of information available. Recall that, as part of this thesis we have collected a combination of color and depth data, it is therefore important to consider, not only which features from each modality to extract, but also how to fuse these features to increase robustness.

2 Related Work

Since it is very difficult to extract features that generalize well across multiple datasets, work has continuously been put into devising features based on the latest knowledge within CV [1, 2]. In recent years, features have been split into two categories; *hand-crafted* features and *deep* features.

Some of the early hand-crafted features for person re-id include extraction of histograms in various color spaces, such as RGB, HSV and YCbCr and fuse those with texture features computed from convolving the images either

with texture filters [3] or local binary patterns (LBP) [4]. Later, more sophisticated hand-crafted features were proposed, such as the salient color name (SCNCD) based features [5], which aims to increase robustness of features against photometric variations. Additionally, it has been common to extract features from patches to increase robustness by capturing local salient information [6–8]. Current state-of-the-art hand-crafted features include the local maximal occurence (LOMO) descriptor proposed in [9], which consists of color features from HSV histograms along with texture features from scale invariant local patterns [10] that are extracted from patches. To further increase feature robustness, max pooling operations are performed across horizontal regions. Finally, the Gaussian of Gaussian descriptor [11, 12] has shown comparable precisions to those of LOMO and uses a hierarchical Gaussian distribution across local regions to capture discriminative information. The principles of LOMO and GOG features are shown in Figure 3.1 (a) and (b), respectively.



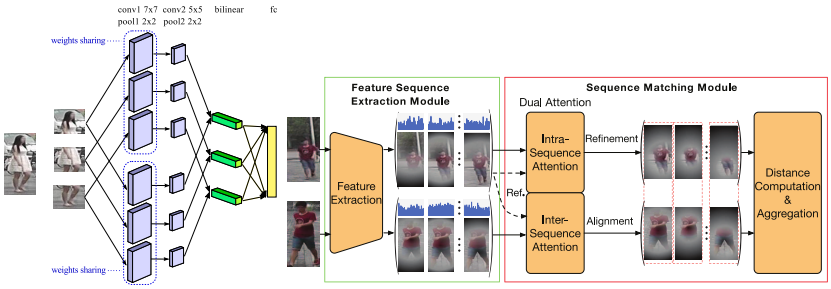
(a) *Image from [9] © 2015 IEEE.*

(b) *Image from [11] © 2016 IEEE.*

Fig. 3.1: The pipelines of state-of-the art hand-crafted features (a) LOMO and (b) GOG.

In 2014, the first works using deep features were presented [13, 14], following that year, the number of published works using deep features for re-id has been only increasing. This was, furthermore, due to publication of larger datasets for re-id, which is required to properly learn deep features. Most commonly is the use of CNNs to train re-id features in an end-to-end fashion by attempting to classify images of persons, based on a person id, and learn from those that were misclassified. The early CNNs horizontally divide images of persons into three or more smaller images to learn body part descriptors that are later fused to a single feature representation [14–18], as shown in Figure 3.2 (a). Later CNNs focus more on local regions, by localizing body joints [19] or keypoints [20]. Based on pioneer work in the deep learning community, CNNs also started to implement spatial attention mechanisms [21] to automatically locate body parts [22–24], or local regions of interest [25–29], as shown in Figure 3.2 (b), to maximize feature discrimination. This has led to several recent datasets almost being solved using deep features [28, 30, 31].

Within RGB-D based re-id, most work is focused on hand-crafted fea-



(a) *Image from [18] © 2017 IEEE.* **(b)** *Image from [26] © 2018 IEEE.*

Fig. 3.2: Example of (a) a part-based CNN model that process upper-body, lower-body and legs independently before fusing features to a single descriptor and (b) a CNN model using spatial attention to capture local semantics.

tures, using either skeleton based features that describe the body shape [32] or using body height and body dimensions as features that are combined with color histograms [33, 34]. More recently, [35] proposed a combination of a CNN and long short term memory (LSTM) network, to model depth across time and fuse depth information with color histograms of only specific body parts. Furthermore, the network implements an attention module to determine the importance of features from subsequent frames.

3 Contributions

In this thesis, features have been devised based on the overhead RGB-D data. Four works have been published centered around feature extraction.

Due to the overhead view, skeleton-based features are not suitable, while body height and ratios are also much dependent on the depth precision to properly work across non-overlapping cameras, a scenario which so far has not been studied. Instead, our proposed solution learn the relevant depth information from CNNs. For both modalities, an AlexNet [36], pretrained on the ImageNet dataset [37], is implemented to learn modality-dependent feature embeddings based on color and depth images. In previous work, features of different modalities are simply concatenated to create a multimodal feature representation [34]. Rather, to find proper correlations between the two modalities, we fuse the features by calculating weighted embeddings, where weights are learned during a training phase [38]. The work resulted in a publication at the *2017 IEEE Conference of the Biometrics Special Interest Group (BIOSIG)* (chapter III.A).

Secondly, while the work in [38] focus mainly on fusing of global feature representations, the work in [39] investigates how to improve the accu-

racy of multimodal deep features. Given the novel research within attention mechanisms, we propose a *Spatial attention* module to capture local semantics within the images. Since early layers of a CNN capture basic color and texture structures, while later layers capture more high-level structures [40], the idea is that such features complement each other well. The idea has been previously explored in re-id by fusing high-level deep features with low-level hand-crafted features [41]. Therefore, for each modality, an attention module is implemented to capture local semantics at different layers of the CNNs, and fuse local features by concatenation. Local features are then fused with global ones to construct multi-level modality-based features for both RGB and depth. These multi-level features are finally fused using a similar strategy as that in [38]. The work has been published at the *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (chapter III.B).

Thirdly, given the experience from [38, 39], an additional module is proposed to increase feature robustness even further. Instead of naively concatenating local features, a weighted average of features is calculated based on learning feature specific weights. This is implemented as a *Layer-wise attention* module, which is combined with the spatial attention module in an architecture that adapts weighting of local features at different abstraction levels based on the input data, and use the weighted local features in construction of multi-level multimodal features [42]. The work has been published in the *IEEE Transactions on Information Security and Forensics (TIFS)* (chapter III.C).

Finally, besides feature fusion of features at different abstraction levels, it is also possible to late fuse features, i.e., conduct re-id classifications for each feature type and fuse those to a single result. To that end, we propose late fusing low-, mid-, and high-level features using two different fusing strategies; rank aggregation, which fuses the ranked lists of matches, and score-level fusion, which fuses the output scores, i.e., calculated distances between a probe and gallery [43]. The work has been published in *IET Biometrics* (chapter III.D) and shows the potential of late fusing features at different abstraction levels, which can be leveraged in future work.

The contributions of feature extraction, thus, can be summarized to the following:

- We propose a CNN architecture that learns and fuses RGB and depth features to a discriminative multimodal feature representation by weighting correlations between the two modalities.
- We propose a spatial attention module to capture local semantics at different abstraction levels of a CNN that are fused with global features to construct multi-level features in case of both RGB and depth. Furthermore, multi-level features are fused to a multi-level multimodal feature representation.

- We propose a layer-wise attention module to dynamically weight and fuse features of local semantics, where the weights of local features are learned through network optimization. The module shows the ability to adapt weights depending the input data.
- Through analyzing the effect of late fusing low-, mid-, and high-level features using two different fusion strategies; rank aggregation based fusion and score-based fusion, we show the potential of late fusing features at different abstraction levels.

References

[1] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[2] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[3] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[4] A. Mignon and F. Jurie, "Pcca: A new approach for distance learning from sparse pairwise constraints," in *Proc. CVPR*. IEEE, 2012, pp. 2666–2672.

[5] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *Proc. ECCV*, 2014, pp. 536–551.

[6] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Proc. CVPR*, 2013, pp. 3586–3593.

[7] ——, "Person re-identification by salience matching," in *Proc. ICCV*. IEEE, 2013, pp. 2528–2535.

[8] X. Liu, M. Song, D. Tao, X. Zhou, C. Chen, and J. Bu, "Semi-supervised coupled dictionary learning for person re-identification," in *Proc. CVPR*, 2014, pp. 3550–3557.

[9] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[10] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. CVPR*, 2010, pp. 1301–1306.

[11] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Proc. CVPR*, 2016, pp. 1363–1372.

[12] ——, "Hierarchical gaussian descriptors with application to person re-identification," *IEEE transactions on pattern analysis and machine intelligence*, 2019, early Access.

[13] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[14] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, 2014, pp. 34–39.

[15] J. Zhu, S. Liao, D. Yi, Z. Lei, and S. Z. Li, "Multi-label cnn based pedestrian attribute learning for soft biometrics," in *Proc. ICB*. IEEE, 2015, pp. 535–540.

[16] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proc. CVPR*, 2016, pp. 1335–1344.

[17] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. CVPR*, 2017, pp. 384–393.

[18] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-region bilinear convolutional neural networks for person re-identification," in *Proc. AVSS*, 2017, pp. 1–6.

[19] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *Proc. CVPR*, 2017, pp. 1077–1085.

[20] M. Saquib Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *Proc. CVPR*, 2018, pp. 420–429.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[22] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[23] L. Zhao, X. Li, Y. Zhuang, and J. Wang, "Deeply-learned part-aligned representations for person re-identification," in *Proc. ICCV*, 2017, pp. 3219–3228.

[24] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proc. CVPR*, 2018, pp. 369–378.

[25] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. CVPR*, 2018, pp. 2285–2294.

[26] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual attention matching network for context-aware feature sequence based person re-identification," in *Proc. CVPR*, 2018, pp. 5363–5372.

[27] C.-P. Tay, S. Roy, and K.-H. Yap, "Aanet: Attribute attention network for person re-identifications," in *Proc. CVPR*, 2019, pp. 7134–7143.

[28] S. Zhou, F. Wang, Z. Huang, and J. Wang, "Discriminative feature learning with consistent attention regularization for person re-identification," in *Proc. CVPR*, 2019, pp. 8040–8049.

[29] G. Chen, C. Lin, L. Ren, J. Lu, and J. Zhou, "Self-critical attention learning for person re-identification," in *Proc. CVPR*, 2019, pp. 9637–9646.

[30] X. Qian, Y. Fu, T. Xiang, Y.-G. Jiang, and X. Xue, "Leader-based multi-scale attention deep architecture for person re-identification," *IEEE transactions on pattern analysis and machine intelligence*, 2019, early Access.

[31] F. Yang, K. Yan, S. Lu, H. Jia, X. Xie, and W. Gao, "Attention driven person re-identification," *Pattern Recognition*, vol. 86, pp. 143–155, 2019.

[32] A. Wu, W.-S. Zheng, and J.-H. Lai, "Robust depth-based person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2588–2603, 2017.

[33] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino, "Re-identification with rgb-d sensors," in *Proc. ECCV*. Springer, 2012, pp. 433–442.

[34] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, "Person re-identification dataset with rgb-d camera in a top-view configuration," in *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 1–11.

[35] N. Karianakis, Z. Liu, Y. Chen, and S. Soatto, "Reinforced temporal attention and split-rate transfer for depth-based person re-identification," in *Proc. ECCV*, 2018, pp. 715–733.

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

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[38] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[39] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "Attention in multimodal neural networks for person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 179–187.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[41] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[42] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[43] A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Enhancing person re-identification by late fusion of low-, mid- and high-level features," *IET Biometrics*, vol. 7, no. 2, pp. 125–135, 2018.

Chapter 4

Practical Re-identification

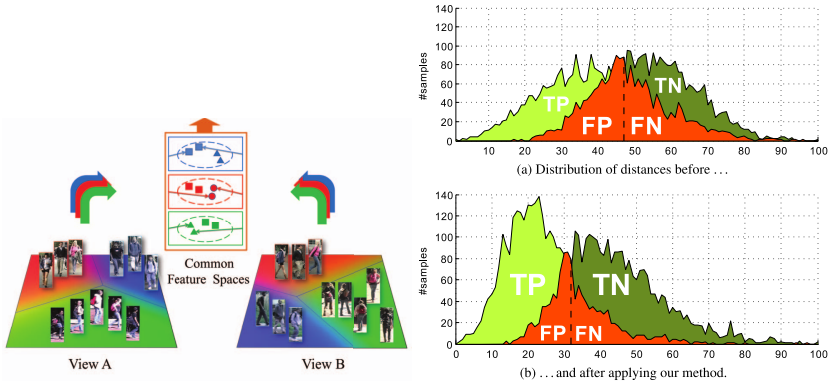
The last part of the thesis aims to bridge the gap between academia and industry by focusing on the practical challenges of running a re-id system. This resolves around transferring learned models to be used across multiple areas of an airport, how to increase the accuracy of a re-id based queue measurement system, and on which platform to run the re-id system.

1 Motivation

To have a re-id system that performs well, it is important to learn features that generalize well across various environments, as was the target in chapter I.3. Learning robust deep features requires a fair amount of labeled data that, typically, would be collected from the environment in which the re-id system is deployed. However, directly transferring a feature extraction model that was trained on data from one environment, to another, often results in significant reductions of precision. Meanwhile, collecting and annotating data every time the re-id system is deployed to a new environment is costly and time consuming. Rather, a re-id model trained in one environment should be transferred to the new one with a minimum loss in precision and data labeling effort.

Instead of transferring the feature extraction model, it is possible to learn a distance metric or projection matrix based on feature pairs of similar and dissimilar persons [1–4]. By learning such relations, distances between features of similar pairs are reduced while those of dissimilar ones are increased, as shown in Figure 4.1. The distance metric can greatly improve the precision of the re-id system and can also be transferred to new environments, however, using the metric directly does not necessarily results in improved precisions. In worst case, transfer of metrics can even lead to a reduction in precision, a phenomena within transfer learning known as *negative transfer*, which occurs

if the distributions of old and new data are very different [5]. To avoid costly data annotation, distance metrics in the new environment could be learned using only a limited number of annotated persons.



(a) *Image from [2] © 2013 IEEE.* **(b)** *Image from [1] © 2012 IEEE.*

Fig. 4.1: (a) Principle of projecting features from non-overlapping views to a common feature space to minimize distances between similar pairs while maximizing those of dissimilar, as shown in (b). In (b), TP: True Positives, FP: False Positives, TN: True Negatives and FN: False Negatives.

Besides transferring of models to new environments, re-id precision can be improved by post processing the output, which is also studied in [6] by late fusion of features at different abstraction levels. Another way is to consider the specific task at hand and how the output should be represented. Since this thesis is based on queue measurements using re-id, the output should resemble the setting of this task to maximize precision of measurements. Specifically, rather than consider a list of ranked persons as the output, only a single person in the gallery should be assigned to each probe.

Deploying a re-id system requires a computing device that can process all steps of the re-id pipeline. To avoid undesirable transfer of image data across a local area network to a server due to, e.g., privacy concerns, the processing should take place at the camera. Therefore, different platforms should be investigated that are able to process image data locally.

2 Related Work

From advances in deep re-id models, more work is put into transferring of knowledge, typically, across datasets [7–9]. Some of the published work consider supervised transfer learning, where a model that was trained on one large dataset is finetuned on a new, likely, smaller dataset to adapt features [7, 10]. Other works adopt generative adversarial networks (GANs)

to translate images from one camera (source domain) to another (target domain), to avoid extensive annotation of data in the target domain [9, 11–14]. Recently, works have been published that perform unsupervised model transfer, either by assigning psuedo labels to images in the target domain [15–18] or by leveraging spatio-temporal information from the target domain [19, 20]. In both cases, a model that was pretrained on the source domain is finetuned in the target domain.

Another important aspect is transfer of knowledge in case of adding a new camera to an existing camera network. This is a problem, which has been studied in only few works [21, 22]. These works consider a pool of source distance metrics computed between camera pairs of the original camera network, from which the most relevant metric is transferred based on similarities between the source and target domains. Similarity between two domains, i.e. cameras, is defined as the distance between data points in each domain, which is calculated using a computed geodestic flow kernel (GFK) [23]. To further improve precision of the best suitable source metric, it is multiplied with the GFK between the domains. However, the works assume the availability of data from the source domain that was used to calculate the source metrics, which is unrealistic in real-world scenarios where data might have to be deleted due to privacy concerns.

In cases of transferring knowledge to new environments, post processing techniques can be used to enhance the re-id precision. In re-id, post processing of the output is concerned around the ranking the persons in the gallery, with respect to the probes, and how the initial ranked list can be altered to improve precision. To that end, re-ranking has been an increasingly popular tool as it considers both the nearest neighbors of the probe, but also how well the probe ranks in relation to the closest gallery images [24–26]. The method has shown to improve re-id precision, even when it is already high [27, 28]. However, the method is mostly suited to the case of forensics, where there might be an operator who observes the ranked lists of likely matches, as shown in Figure 4.2 (a), hence, it does not consider the case of assigning only a single gallery person to each probe, as shown in Figure 4.2 (b).

While maximizing re-id precision is, of course, the primary goal, from a practical point of view it is also important to consider the complexity of the feature extractor and find an optimal trade-off between running time and precision based on available hardware. A few recent works also consider complexity in development of feature extractors [30–33], either by having CNN architectures that are based on few optimized layers [33, 34] or by distilling knowledge from larger and more complex CNNs to simple and faster ones [35, 36]. Even though, the works show important aspects of deploying re-id models, running time is not evaluated on hardware, which is able to run locally at the camera.

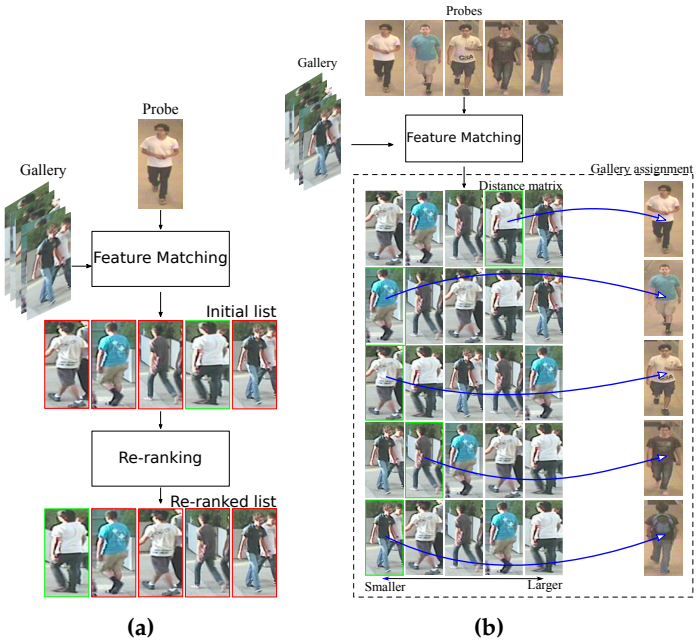


Fig. 4.2: (a) Re-ranking the initial list for each probe with the probability of having the same gallery person assigned to multiple probes, and (b) one-to-one assignment with the constraint of only matching each gallery person to a single probe. *(b) is also used in [29].*

3 Contributions

Due to the limited work within knowledge transfer from existing cameras to a newly introduced camera in a camera network, we investigate how transferring of knowledge from existing cameras to a new camera can be accomplished with limited labeling efforts and without the use of old data from the existing camera network. Since, with limited data, it is not possible to properly finetune deep features to a new environment, the transfer should happen from source distance metrics. *Hypothesis transfer learning* (HTL) [37, 38] is a type of transfer learning, which aims to transfer knowledge from existing learned metrics with only limited labeling effort in the newly installed camera. The method combines distance metrics computed between camera pairs in source domains (source metrics) with limited labeled data in the target domain, to calculate a new and optimized metric between the target and a source domain. HTL has been mostly applied from linear classifiers in an SVM framework [39, 40], with the exception of a single work [41], which considers transfer of a single source metric defined as a covariance matrix. Furthermore, HTL allows transferring of knowledge from multiple source metrics, which has been explored only for linear classifiers [39], by calculat-

ing a weighted average of source metrics based on an optimization scheme that jointly optimizes the target metric and source metric weights. As a result, HTL is proposed in [42] to transfer knowledge from multiple source metrics, defined as covariance matrices, using only limited data in the expanded camera network after introduction of a new camera, as shown in Figure 4.3. Experimental results show the capability of the method to adapt newly introduced cameras with precisions that are comparable to supervised distance metric learning with sufficient training data, using source metrics and only limited labeled data. Furthermore, from both a theoretical analysis and experimental results, we show that the method minimizes the risk of negative transfer due to weighting the source metrics. The work has been submitted for the *2020 IEEE Conference of Computer Vision and Pattern Recognition (CVPR)* (chapter IV.E) and is currently under review.

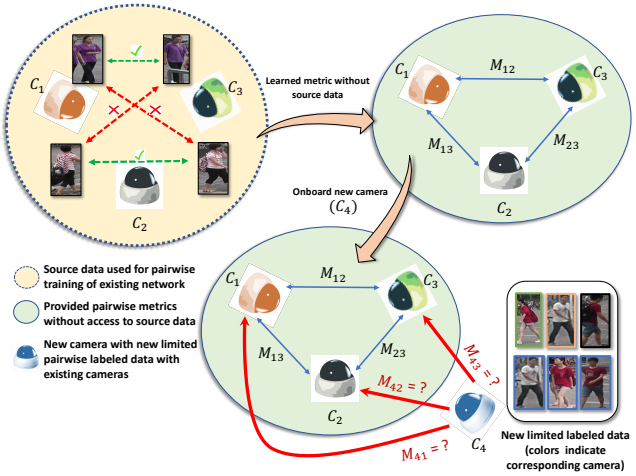


Fig. 4.3: Principle of applying HTL to learn pairwise metrics between a newly introduced camera and existing cameras in a camera network using source metrics that was learned in between the existing camera pairs and only limited labeled data from the expanded network. *Image from [42].*

To further increase re-id precision, e.g, in cases where HTL has been applied to transfer models to a newly installed camera, a post processing technique should be applied, which considers the re-id context. To measure queue times using re-id, gallery persons should be assigned to each probe. Usually, this would be done by assigning the most similar person, i.e., the one with the shortest distance. But doing so, there is a probability of assigning the same gallery person to multiple probes, without the knowledge of which of the matches that is correct. Instead, a gallery person should be assigned only to a single probe, an assumption which is valid only in case of the close-world re-id setting. In queues where the path between entrance and exit is fixed, this is a valid assumption. Furthermore, additional information

should be considered to constraint the possible matches, such as the entrance and exit order of the passengers, which will assign a subset of passengers in the queue higher probabilities to be possible matches. To decide the optimal set of matches, we propose an optimization method, which applies the Hungarian algorithm [43] to minimize the total distance between probes and gallery, and assign probes unique id's. We consider assignment using the k most likely matches to each probe, where different values of k are evaluated to maximize re-id precision. The work shows an increase of precision and recall compared to naively assigning the most similar match, which results in median queue time measurements that deviate only a few percentages compared to ground truth. The paper is on-going work and is in this thesis included as a technical paper (chapter IV.F).

The first step towards deployment of a re-id system is to investigate and identify possible candidate platforms. In recent years, several edge platforms have been brought onto the market by big companies, such as the NVIDIA Jetson boards [44], the Intel Neural Compute Sticks (NCS) [45] and Google Coral [46]. These platforms all have the option to optimize and deploy models with limited power usage while maintaining decent computational power. As a step towards identifying the optimal platform for a re-id system, we evaluate selected edge platforms within the common CV tasks of object classification, object detection and semantic segmentation [47] (chapter IV.G). For each task, inference time of the platforms are compared across models of different complexity and with different batch sizes. Additionally, retail price of the platforms is considered to aid the optimal platform/model selection based on requirements on both budget and inference speed. Finally, timings of different DNN operations within each platform are analyzed to identify operations that should be the main target of optimization to increase inference time, based on which platform is acquired. The work is submitted to the journal of *Neural Computing and Applications* and is currently under review.

The contributions of practical re-identification can summarized to the following:

- We propose an HTL optimization algorithm to transfer knowledge from multiple source metrics trained between camera pairs of an existing camera network, to a newly introduced camera where only limited labeled data is available.
- We provide theoretical and experimental analysis that validates the ability of the proposed transfer learning to minimize negative transfer while maintaining high precision.
- We propose an algorithm to improve re-id precision by assign probe-gallery pairs based on using the Hungarian algorithm to minimize the total distances.

- We evaluate and compare selected edge platforms across models of different complexity on three CV tasks, to aid optimal platform/model selection, and identify DNN operations that mostly affect inference time based on platform.

References

[1] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[2] W. Li and X. Wang, "Locally aligned feature transforms across views," in *Proc. CVPR*, 2013, pp. 3594–3601.

[3] S. Li, M. Shao, and Y. Fu, "Cross-view projective dictionary learning for person re-identification," in *Proc. AAAI*. AAAI Press, 2015, pp. 2155–2161.

[4] S. Liao and S. Z. Li, "Efficient psd constrained asymmetric metric learning for person re-identification," in *Proc. ICCV*, 2015, pp. 3685–3693.

[5] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proc. CVPR*, 2019, pp. 11 293–11 302.

[6] A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Enhancing person re-identification by late fusion of low-, mid- and high-level features," *IET Biometrics*, vol. 7, no. 2, pp. 125–135, 2018.

[7] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proc. CVPR*, 2016, pp. 1249–1258.

[8] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proc. CVPR*, 2018, pp. 2275–2284.

[9] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proc. CVPR*, June 2018, pp. 994–1003.

[10] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," *arXiv preprint arXiv:1611.05244*, 2016.

[11] S. Bak, P. Carr, and J.-F. Lalonde, "Domain adaptation through synthesis for unsupervised person re-identification," in *Proc. ECCV*, 2018, pp. 189–205.

[12] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person trasfer gan to bridge domain gap for person re-identification," in *Proc. CVPR*, 2018, pp. 79–88.

[13] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *Proc. CVPR*, 2018, pp. 5157–5166.

[14] Y. Chen, X. Zhu, and S. Gong, "Instance-guided context rendering for cross-domain person re-identification," in *Proc. CVPR*, 2019, pp. 232–242.

[15] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *Proc. ICCV*, 2017, pp. 994–1002.

[16] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 4, p. 83, 2018.

[17] H. Chen, Y. Wang, Y. Shi, K. Yan, M. Geng, Y. Tian, and T. Xiang, "Deep transfer learning for person re-identification," in *Proc. BigMM*. IEEE, 2018, pp. 1–5.

[18] H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised person re-identification by soft multilabel learning," in *Proc. CVPR*, 2019, pp. 2148–2157.

[19] J. Lv, W. Chen, Q. Li, and C. Yang, "Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns," in *Proc. CVPR*, 2018, pp. 7948–7956.

[20] M. Li, X. Zhu, and S. Gong, "Unsupervised tracklet person re-identification," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[21] R. Panda, A. Bhuiyan, V. Murino, and A. K. Roy-Chowdhury, "Unsupervised adaptive re-identification in open world dynamic camera networks," in *Proc. CVPR*, 2017, pp. 7054–7063.

[22] ——, "Adaptation of person re-identification models for on-boarding new camera (s)," *Pattern Recognition*, vol. 96, p. 106991, 2019.

[23] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. CVPR*. IEEE, 2012, pp. 2066–2073.

[24] Q. Leng, R. Hu, C. Liang, Y. Wang, and J. Chen, "Person re-identification with content and context re-ranking," *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 6989–7014, 2015.

[25] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. CVPR*, 2017, pp. 1318–1327.

[26] M. Saquib Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *Proc. CVPR*, 2018, pp. 420–429.

[27] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3037–3045, 2019.

[28] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proc. CVPR*, 2018, pp. 1062–1071.

[29] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "One-to-one person re-identification in a queue," Aalborg University, Tech. Rep., 2019.

[30] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[31] J. Zhu, H. Zeng, S. Liao, Z. Lei, C. Cai, and L. Zheng, "Deep hybrid similarity learning for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3183–3193, 2017.

[32] M. Ye, A. J. Ma, L. Zheng, J. Li, and P. C. Yuen, "Dynamic label graph matching for unsupervised video re-identification," in *Proc. ICCV*, 2017, pp. 5142–5150.

[33] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. CVPR*, 2018, pp. 2285–2294.

[34] Y. Guo and N.-M. Cheung, "Efficient and deep person re-identification using multi-level similarity," in *Proc. CVPR*, 2018, pp. 2335–2344.

[35] A. Wu, W.-S. Zheng, X. Guo, and J.-H. Lai, "Distilled person re-identification: Towards a more scalable system," in *Proc. CVPR*, 2019, pp. 1187–1196.

[36] I. Ruiz, B. Raducanu, R. Mehta, and J. Amores, "Optimizing speed/accuracy trade-off for person re-identification via knowledge distillation," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103309, 2020.

[37] I. Kuzborskij and F. Orabona, "Stability and hypothesis transfer learning," in *Proc. ICML*, 2013, pp. 942–950.

[38] S. S. Du, J. Koushik, A. Singh, and B. Póczos, "Hypothesis transfer learning via transformation functions," in *Proc. NIPS*, 2017, pp. 574–584.

[39] Y.-X. Wang and M. Hebert, "Learning by transferring from unsupervised universal sources," in *Proc. AAAI*, 2016, pp. 2187–2193.

[40] D. Benavides-Prado, Y. S. Koh, and P. Riddle, "Selective hypothesis transfer for lifelong learning," in *Proc. IJCNN*. IEEE, 2019, pp. 1–10.

[41] M. Perrot and A. Habrard, "A theoretical analysis of metric hypothesis transfer learning," in *Proc. ICML*, 2015, pp. 1708–1717.

[42] S. M. Ahmed, A. R. Lejbølle, R. Panda, and A. K. Roy-Chowdhury, "Camera on-boarding for person re-identification using hypothesis transfer learning," November 2019, under review for the 2020 IEEE Conference on Computer Vision and Pattern Recognition.

[43] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[44] N. Developer. (2019) Nvidia jetson. https://developer.nvidia.com/embedded/develop/hardware. NVIDIA. Accessed: November 28, 2019.

[45] Intel. (2019, September) Intel neural compute stick 2. https://software.intel.com/en-us/neural-compute-stick. Accessed: November 28, 2019.

[46] Google. (2019) Coral. https://coral.withgoogle.com/. Google. Accessed: November 28, 2019.

[47] A. R. Lejbølle, C. B. Rasmussen, K. Nasrollahi, and T. B. Moeslund, "Evaluation of edge platforms for deep learning in computer vision," under review for the journal of Neural Computing and Applications.

Chapter 5

Summary

This thesis is based on vision-based person re-id in a queue by comparing features that are captured by a camera at the queue exit to features that are captured by a camera at the queue entrance. While re-id based queue measurements can be applied to queues anywhere in the world, focus of the thesis is on queues in airports. Current camera based solutions are costly in low-ceiling areas and WiFi/BT solutions are challenged by changes in the OS's of both Android and iOS, however, the proposed solution requires a minimum of only two cameras, one at the entrance and one at the exit.

As the re-id pipeline covers many large tasks, focus of this thesis is only on the most relevant ones, one of which covers re-id on a more practical level. The following tasks, thus, are studied:

- Data Acquisition
- Feature Extraction
- Practical Re-identification

Most current re-id datasets were collected from a, primarily, horizontal viewpoint, which increases the probability of occlusions. Furthermore, the horizontal viewpoint allows the cameras to cover larger areas where persons move around unconstrained, which is not the case in queues where passengers are constraint to follow a predefined maze. As a result, data were collected from an overhead viewpoint. While an overhead viewpoint decreases the probability of occlusions in-between persons or between persons and objects, it does increase the probability of self-occlusion. Therefore, the data were collected from additional modalities, more specifically, the depth modality, which is an obvious choice given the overhead viewpoint. This resulted in collection of two novel overhead RGB-D datasets, one from a university canteen (OPR) and another from an airport (QPR). While previous

proposed overhead RGB-D datasets for re-id were collected either in a very controlled environment or with very few persons, the collected datasets are targeted re-id in a queue, additionally, the number of persons in the datasets are higher. Furthermore, the second QPR dataset was collected from two non-overlapping cameras, in contrast to the previous published datasets that were collected from only a single camera.

Based on the collected data, novel features have been devised. Due to the availability of both RGB and depth images, we propose learning a combination of RGB and depth features through training a CNN. Rather than just concatenating RGB and depth features to a single feature representation, the correlations between the two modalities should be learned. Thus, we propose an architecture where modality-dependent features embeddings are produced by independent CNNs, and fused as a weighted sums of features, where weights are learned through optimizing the entire network. Additionally, to make modality-dependent features more discriminative, a spatial attention module is proposed, which captures local semantics at different abstraction levels of a CNN, which are concatenated with global feature descriptors to make up a multi-level feature descriptor. Finally, instead of naively concatenating features at different abstraction levels and potentially weight noisy features equally to discriminative ones, a layer-wise attention module is proposed to dynamically weight features and produce a local feature descriptor, which is calculated as the weighted average of multiple local features. The combination of spatial- and layer-wise attention is not only capable of capturing important local semantics, but also weight those depending on the input data.

Besides calculating a weighted average of features at different abstraction levels, it is also possible to late fuse the outputs of applying classification on each local feature, independently. This is analyzed by using the late fusion strategies of rank-based fusion and score-based fusion. The results show a potential in late fusing features at different abstraction levels and could in future works be transferred to features from different layers of a CNN.

Since only little work has been put into re-id from a more practical view, work is conducted with the focus of challenges in deploying a re-id system. First, the problem of extending a current camera network by introducing new cameras is explored. As previous works often requires large quantities of labeled data when transferring knowledge between cameras, an HTL based optimization algorithm is proposed that consider source metrics from existing cameras and only requires little labeling effort in the newly introduced camera to maintain a good re-id precision. The algorithm, further, makes it possible to transfer knowledge from multiple previous cameras without the risk of negative transfer due to weighting of source metrics.

Besides the ability to transfer knowledge to new cameras with little labeling effort, an algorithm is also proposed that post process the re-id outputs

based on the context of queue measurements. Since in this context, there are no operators to monitor the ranked lists that are output, there is no way to tell which of the most similar matches to the probes that are correct. Therefore, the Hungarian algorithm is used to assign a set of gallery-probe pairs, which minimizes the total distance.

Having a re-id system that performs well is of most importance, however, to avoid transfer of large quantities of image data across a local network, the system should be able to process data locally at the camera. This requires the use of an edge platform, which is able to run at low power consumption while maintaining computational power. Therefore, we evaluate selected edge platforms, which can optimize and deploy DNN models at the edge, across three common CV. For each task, inference time of each platform is measured across models of different complexities and precisions. By also including retail price of the platforms, we analyze and compare the performance with respect to the price between the platforms and models, to aid the selection of an optimal platform/model combination based on requirements on speed and budget. Finally, from analyzing the computation time of different DNN operations across platforms, we identify for each platform the operations that should be of greatest focus when optimizing the models.

All contributions made from this Ph.D. can, thus, be summarized to the following:

Data Acquisition

- We collected two overhead RGB-D based datasets using high-resolution cameras to capture fine-grained details of both color, texture and depth. Both datasets were collected in uncontrolled environments.
- Through experiments in [1–3] we show that OPR is a more complex and difficult dataset to solve compared to previously published TVPR and DPI-T datasets.
- In [4] we use QPR to evaluate queue time measurements using vision-based re-identification.

Feature Extraction

- We propose a CNN architecture that learns and fuses RGB and depth features to a discriminative multimodal feature representation by weighting correlations between the two modalities.
- We propose a spatial attention module to capture local semantics at different abstraction levels of a CNN that are fused with global features to

construct multi-level features in case of both RGB and depth. Furthermore, multi-level features are fused to a multi-level multimodal feature representation.

- We propose a layer-wise attention module to dynamically weight and fuse features of local semantics, where the weights of local features are learned through network optimization. The module shows the ability to adapt weights depending the input data.
- Through analyzing the effect of late fusing low-, mid-, and high-level features using two different fusion strategies; rank aggregation based fusion and score-based fusion, we show the potential of late fusing features at different abstraction levels.

Practical Re-identification

- We propose an HTL optimization algorithm to transfer knowledge from multiple source metrics trained between camera pairs of an existing camera network, to a newly introduced camera where only limited labeled data is available.
- We provide theoretical and experimental analysis that validates the ability of the proposed transfer learning to minimize negative transfer while maintaining high precision.
- We propose an algorithm to improve re-id precision by assign probe-gallery pairs based on using the Hungarian algorithm to minimize the total distances.
- We evaluate and compare selected edge platforms across models of different complexity on three CV tasks, to aid optimal platform/model selection, and identify DNN operations that mostly affect inference time based on platform.

References

[1] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[2] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "Attention in multimodal neural networks for person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 179–187.

[3] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[4] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "One-to-one person re-identification in a queue," Aalborg University, Tech. Rep., 2019.

Part II

Data Acquisition

Chapter 1

Introduction

As previous published datasets have not considered re-id in a queue, the very first part of the project deals with data acquisition from an overhead viewpoint. To quickly begin devising novel features, data were collected in a local environment using a single camera, where queues would be naturally formed. The resulting dataset was collected in a university canteen where queues occur during lunch time. Furthermore, due to the layout of the canteen, a single camera could be used to capture persons both when standing in queue and when leaving the canteen area. The dataset, thus, is simple, however, suitable in case of devising relevant features. Data collection of this dataset, OPR, will be further described in chapter II.2.

Later in the project period, data were collected from an actual airport queue, in order to properly evaluate the novel features. The goal was to target a queue with a single entrance and exit at non-overlapping locations. Data were collected from an immigration area and contain both images and timestamps. The resulting QPR dataset can, thus, be used not only to evaluate features but also queue time measurements from re-id, by comparing to ground truth queue times. The dataset will be further described in chapter II.3.

1 Choice of 3D Camera

Due to collection of both RGB and depth data, a camera had to be used that with the option of producing depth maps. To that end, several options were available based on different technology; active stereo, which include a variety of technologies, or passive stereo using two regular RGB cameras.

Active stereo is based on projecting a pattern and compute depth from correspondences between points in the pattern. Broadly, it can be categorized

into three categories; (1) cameras that emit a structured near infrared light (SL), i.e., pattern, from a laser projector, and compute depth from correspondence between points in the pattern [1, 2], (2) cameras that emit a light pattern and measures the time from the light hits an object till it returns, also known as time of flight (ToF) [3, 4], (3) and cameras that emit an infrared pattern and use a pair of infrared sensors to compute the depth from correspondences between distinguishable points in the patterns [5, 6].

Passive stereo cameras do not project patterns or emit light, rather, they compute depth from correspondences of distinguishable image points between two RGB cameras that are placed side-by-side. The distance between the cameras is referred to as the *baseline* and determines the distance from which depth can be perceived.

An overview of widely-used stereo cameras is shown in Table 1.1. To maximize coverage in low ceiling areas, a large field of view (FoV) is required, additionally, the camera should work in heights of at least 4 m. A couple options were available, such as the Orbbec Pro, ZED and KinectV2[1]. The Orbbec Pro can reach up to 8 m, but has a narrow FoV and the use of SL can cause issues in brightly lid areas. Even though, the FoV is larger in case of the KinectV2, the lighting issue remains. In case of the ZED camera, depth is computed from the two RGB images using algorithms that are more processing heavy than those used in case of active stereo cameras. Nonetheless, the FoV of the ZED is much larger while the resulting resolution of the depth maps is also much higher. As a result, we chose the ZED camera for the data collection. The camera was used in case of both OPR and QPR.

2 Camera Calibration

To have very precise depth maps, it is important to properly calibrate the cameras to alleviate lens distortion, which is the effect of having images that curve at the corners. Since a passive stereo camera was used to collect data, calibration had to be done independently for each RGB camera, followed by at stereo calibration step to align images captured by each camera.

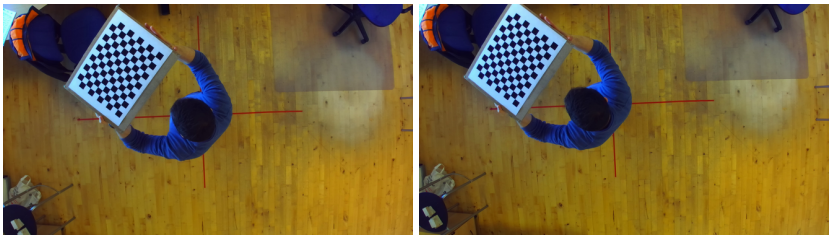
We calibrated the cameras using images of a checkerboard printed on a solid wood board, as shown in Figure 1.1. The checkerboard has a height and width of 12 and 9 squares, respectively, where each square has a size of 2.7×2.7 cm.

To calibrate the first camera, which was used to collect both OPR and QPR, 99 images were captured with various positions and orientations for

[1]Please note that the two Realsense cameras and Kinect Azure were not yet launched at the time of the data collection.

	Technology	Depth resolution	RGB resolution	Field of View [°]	Range [m]
ZED [7]	Passive stereo	2208x1242	2208x1242	H: 90 V: 60 D: 110	0.5-20
KinectV2 [3]	Time of Flight	512x424	1920x1080	H: 70 V: 60	0.5-4.5
XtionPro Live [1]	Structured light	640x480	1280x1024	H: 58 V: 45 D: 70	0.8-3.5
Orbbec Pro [2]	Structured light	640x480	1280x720	H: 60 V: 49.5 D: 73	0.6-8.0
Realsense D415 [5]	Active IR stereo	1280x720	1920x1080	H: 65 V: 40 D: 72	0.16-10
Realsense D435 [6]	Active IR stereo	1280x720	1920x1080	H: 87.3 V: 58 D: 95	0.1-10
Azure Kinect [4]	Time of Flight	1024x1024	4096x3072	H: 120 V: 120	0.25-2.21

Table 1.1: Overview of common stereo cameras. Reported resolutions are maximum supported. H: Horizontal, V: Vertical and D: Diagonal.



(a) **(b)**

Fig. 1.1: Examples of (a) left and (b) right image from the ZED camera used for calibration.

left and right view, respectively. In calibration of the second camera, which we used only to collect the QPR dataset, 67 images were captured. For OPR, the calibration targeted full hd (FHD) resolution (1920×1080), while calibration of both cameras targeted 2k resolution (2208×1242) in case of QPR. In all cases, we calibrated the cameras using precoded scripts[2], which are based on standard OpenCV functions [8]. First, the findChessboardCorners function is used to locate the checkerboard corners, followed by calibrateCamera to compute camera intrinsics and distortion coefficients based on those. The outputs are a 3×3 matrix containing the camera intrinsics along with a 1×5 vector containing distortion coefficients. Similarly, to compute the extrinsics between the two cameras, we run StereoCalibrate, which computes a rotation matrix along with a translation vector, based on identified checkerboard corners and camera-wise intrinsics. The rotation matrix and translation vec-

[2]Available from: `https://github.com/sourishg/stereo-calibration/`

tor relate the position of the two cameras. The camera extrinsics are then used, along with the camera intrinsics, to compute a reprojection matrix using the STEREORECTIFY function, which is later used to calculate depth maps. We verify calibrations by running COMPUTEREPROJECTIONERRORS to calculate the average absolute norm between the corner points found from FINDCHESSBOARDCORNERS and those from projecting object points to image points using the computed camera intrinsics. In case of stereo calibration, STEREOCALIBRATE outputs the average re-projection error based on both views. In either cases, the re-projection errors in pixels (px) should be as close to zero as possible, with optimal values being < 0.5 px. The re-projection errors for the two calibrated ZED cameras are shown in Table 1.2.

	Left camera	**Right camera**	**Stereo**
ZED 1 (FHD)	0.187	0.186	0.221
ZED 1 (2k)	0.292	0.289	0.313
ZED 2 (2k)	0.227	0.189	0.240

Table 1.2: Re-projection errors (in px) from right, left and stereo calibration. Values < 0.5 px are good values.

3 Depth Calculation

Since passive stereo cameras use images from the left and right sensors to calculate a depth map, first the images have to be aligned. This is done through undistortion and rectification of left and right image, respectively, which fixes lens distortion and geometrically transforms the image. In OpenCV, this is done by calculating joint undistortion and rectification transformation matrices for each camera, independently, using the INITUNDISTORTRECTIFYMAP. Next, the images are transformed using the transformation matrices in the REMAP function, where the values of each pixel in the transformed image are based on interpolation of pixel values at the same coordinate in the original image. The principle is shown in Figure 1.2.

Before calculating the depth map itself, a disparity map is calculated from matching points in the left and right image. When calculating the disparity maps, the distance in pixels between a point in the left image and its corresponding location in the right image is calculated. OpenCV offers a couple of standard solutions for stereo matching using a pair of rectified images, including the *block matching* (BM) and *semi global block matching* (SGBM) algorithms that are based on similar ideas.

BM is a relatively simple matching technique, which takes a block around the point of interest in the left image and matches is with blocks at same horizontal location in the right image. Since the right image is shifted relatively

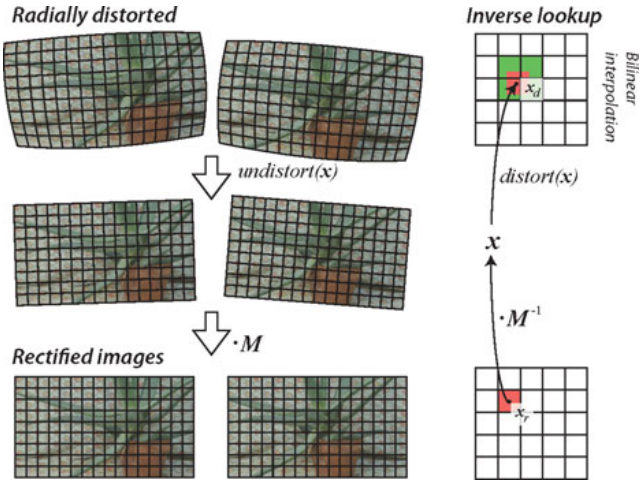


Fig. 1.2: Principle of fixing lens distortion and rectifying left and right images. *Image from [9].*

to the left, a horizontal search is performed within a bounded area of the point of interest, both to the left and the right. At each location, a sum of absolute differences (SAD) is calculated between the block from the left image and current block in the search region, to find the most similar one. The matching block in the right image is the one were the SAD is lowest and the corresponding disparity value is the distance between the center locations of the left and right block, respectively. The principle is shown in Figure 1.3. The size of the block determines the coarseness of the disparity map, a larger block size results in less noisy disparity maps, however, at the cost of capturing less details. To speed up calculations, it is also possible to first convert images to gray scale. This results in the SAD only being calcuated across a single channel.

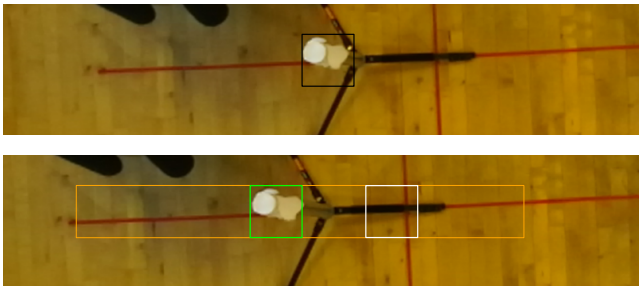


Fig. 1.3: For the black box in the left (top) image, search is performed in the orange rectangle in the right (bottom) image. For each location, indicated by the white box, an SAD is calculated. The green box indicate the closest match and the disparity is calculated as the horizontal distance in pixels between center points of the black and green boxes.

SGBM is merely an extension of BM. The algorithm also performs block search, but differently from BM, it searches in up to eight directions; horizontally, vertically and diagonally. SGBM is a computationally heavy stereo matching algorithm compared to BM. To speed up calculations, an option is to do the block search on downsized images and, optionally, upscale the disparity map afterwards.

Since our initial focus is to maximize re-id precision, we calculate disparity maps using the SGBM algorithm as it produces more precise ones, which is important in order to extract discriminative depth features. To enhance the quality of depth maps, we further apply a filtering step using the weighed least square filter (WLS), which smoothen the image to make the background more uniform and eliminate noise. On an Intel i7-6700HQ CPU @ 2.60GHz, SGBM and WLS take on average 0.136 s and 0.103 s, respectively, to process a single pair of RGB images (numbers from [10]). Afterwards, the disparity map is used to calculate the depth map using the reprojectImageTo3D function in OpenCV, which reprojects the disparity map to 3D space using the reprojection matrix, which is computed upon stereo calibration. In the new depth map, each pixel contained (x,y,z) coordinates, thus, we extract the third channel containing z-coordinates as this provide the relevant depth information.

In the following chapters, each dataset will be presented, including the set-up, software used for collection and annotation, and dataset statistics.

Chapter 2

University Canteen Dataset

As mentioned in chapter II.1, the OPR was collected as basic queue-based re-id dataset with feature devising in mind. Thus, the goal has not been to evaluate queue measurements using re-id, rather, evaluate feature performance. The dataset was collected during a two hour period at midday, when the number of persons in the canteen would peak.

1 Hardware set-up

A single ZED camera (ZED 1) was used to collect this dataset. To have the camera point downwards, it was attached to a wooden board, which was placed on a cable tray below the ceiling. The camera was oriented to capture persons both when entering and leaving the canteen. An overview of the camera coverage seen from above is shown in Figure 2.1 (a), the horizontal view of the set-up is shown in Figure 2.1 (b), while Figure 2.1 (c) shows the actual view of the camera. Additionally, Figure 2.2 shows an example of a depth image where the queue appears in the right side and persons who exit the queue reappears in the left side.

2 Data Collection and Annotation

The data were collected using the ZED Explorer tool, which is a part of the ZED SDK [7]. The tool has the option of saving video both in raw format or using image compression, such as PNG or H.264. Furthermore, it is also possible to change camera settings, such as exposure, contrast and brightness. To minimize motion blur, and eliminate noise in the colors, we reduced exposure as much as possible, without making the images too dark. Finally, data were recorded in FHD resolution at a frame rate of 30 images per second.

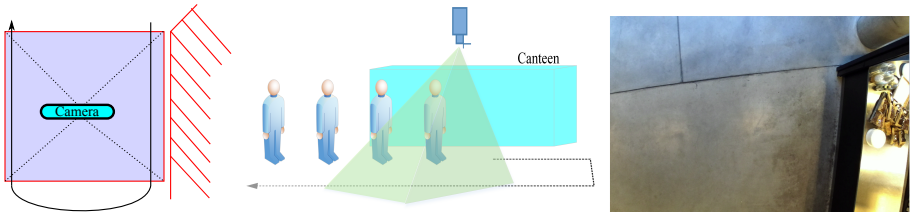


Fig. 2.1: (a) Overhead coverage of the camera. The red hatched area indicates the canteen while the dotted lines and light blue area indicate the camera coverage. The arrows shows the walking path of persons in the queue. (b) Horizontal view of the set-up, where the canteen is represented as a solid box and the walking path of the persons in the queue is indicated by the black dotted line.(c) image from the camera view, where the canteen is seen in the right side.

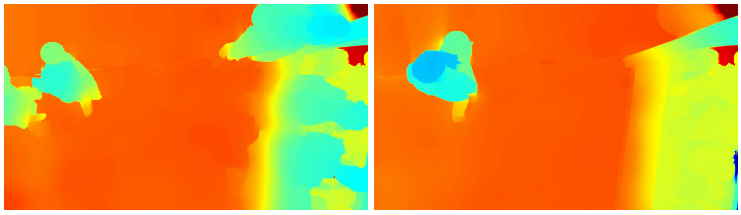


Fig. 2.2: (a) Example of a depth map with a queue in the right side and reappearing persons in the left, (b) and an example of a single person reappearing in the left side. Please note that the depth maps are converted to color images by applying a JET color map. *Images from [10].*

Since focus of thesis is not on person detection, data were manually annotated. Due to the uncontrolled environment, some of persons did not appear twice within the camera view, as a result, we annotated only those that reappeared. We annotated the data using the video annotation tool from Irvine California (VATIC)[1] [11], which is free, easy to use and has docker integration. An example screen shot of the tool is shown in Figure 2.3. The tool can be used to draw bounding boxes around predefined objects in each frame, however, rather than drawing boxes in each frame, the tool also applies interpolation when a bounding boxes is drawn and moved a few frames later. Moreover, it is possible to provide flags, based on whether objects are occluded or outside the frame. As a result, it is possible to assign an object the same id once it reappears in the frame.

[1] Available from: `http://www.cs.columbia.edu/~vondrick/vatic/`

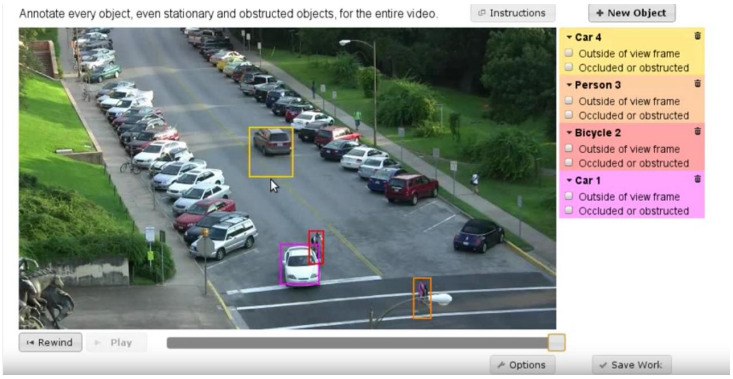


Fig. 2.3: VATIC annotation tool [11]

3 Data Statistics

An overview of the dataset statistics is shown in Table 2.1. In total, 64 persons have been annotated, which have resulted in 44,626 annotated bounding boxes in case of both RGB and depth. On average, each person is represented by almost 700 bounding boxes.

	Entrance	**Exit**	**Total**
Persons	64	64	64
Bounding boxes	41,883	2743	44,626
Avg. bounding boxes	654.42	42.86	697.28

Table 2.1: Overview of OPR dataset.

In addition, Figure 2.4 shows the distribution of bounding boxes per person. While there are some outliers with over 1500 bounding boxes, the majority of persons lie in the range of 100 to 500 annotated bounding boxes.

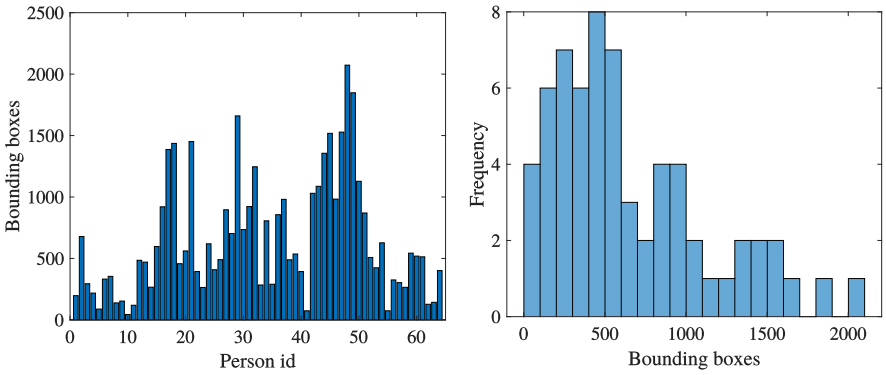


Fig. 2.4: (a) Number of bounding boxes per person and (b) frequency in number of bounding boxes in bins of 100.

Chapter 3

Airport Dataset

The OPR dataset from the university canteen was collected with feature devising in mind. The QPR dataset, on the other hand, was collected with the purpose of also evaluating queue time measurements using vision-based re-id. As a result, this dataset was collected at an airport immigration area, in a setting similar to that in, e.g., an airport security check. The dataset was collected in the morning during arrival of a flight where passengers passed through immigration.

1 Hardware set-up

Due to non-overlapping entry and exit points, two ZED cameras were used. Similarly to section II.2.1, mounts were built to have the ZED cameras point downwards. Due to the ceiling at the entrance consisting of removable tiles, a mount was built, which could rest on the tracks holding the tiles. The mount consists of a long cable tray to fixate the camera (ZED 1), as shown in Figure 3.1 (a) and (b). To make more features visible, and avoid motion blur, two extra light sources were placed next to the camera. With additional light sources, a lower exposure time could be used during recording. To have clear and distinguishable colors, we chose light sources that have a high color rendering index (CRI) and color temperatures close to daylight (5000 K) [12]. For this purpose, two LITE Panel light ECO 60 were used, which are square LED lights with color temperatures of 4000 K and CRI values of 92 [13]. Due to the size of the LED panels, they simply replaced the ceiling tiles. The lights were placed right before and after the camera, as shown in Figure 3.1 (b).

At the exit, the camera (ZED 2) was placed at a height from which it could cover the entire exit. To heighten the placement of the camera, it was mounted on a wooden stick, which was attached to a horizontal wooden bar to stabilize the camera. The camera mount and the placement of the camera

at the queue exit are shown in Figure 3.1 (c) and (d), respectively.

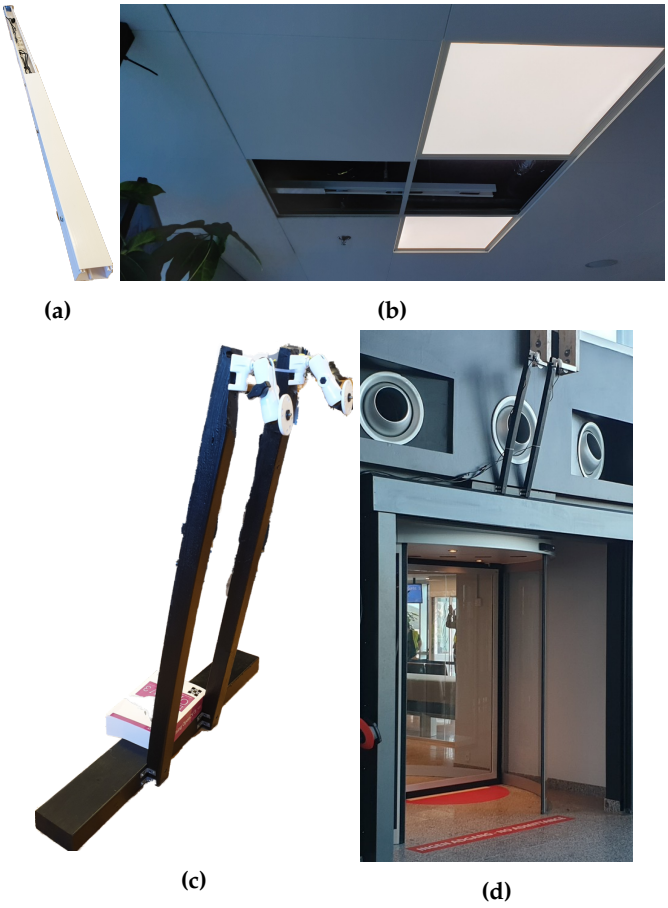


Fig. 3.1: (a) The camera mount used at the entrance and (b) the corresponding placement in the ceiling along with the two extra LED light sources. (c) The camera mount used at the exit and (d) the placement of the mount above the exit.

In contrast to the OPR dataset, persons in this dataset could not move around unconstrained, rather, they had to follow a maze structure, defined by queue barriers. An overview of the maze is shown in Figure 3.2, including the camera coverage at the entrance and exit. From entering the queue to reaching passport control, the queue follows a single lane. At the passport control, however, the queue splits into two lanes, one for each of the operators. This cause queue times to be different between persons, and denies the assumption that the order of which persons entered the queue is the same at the exit.

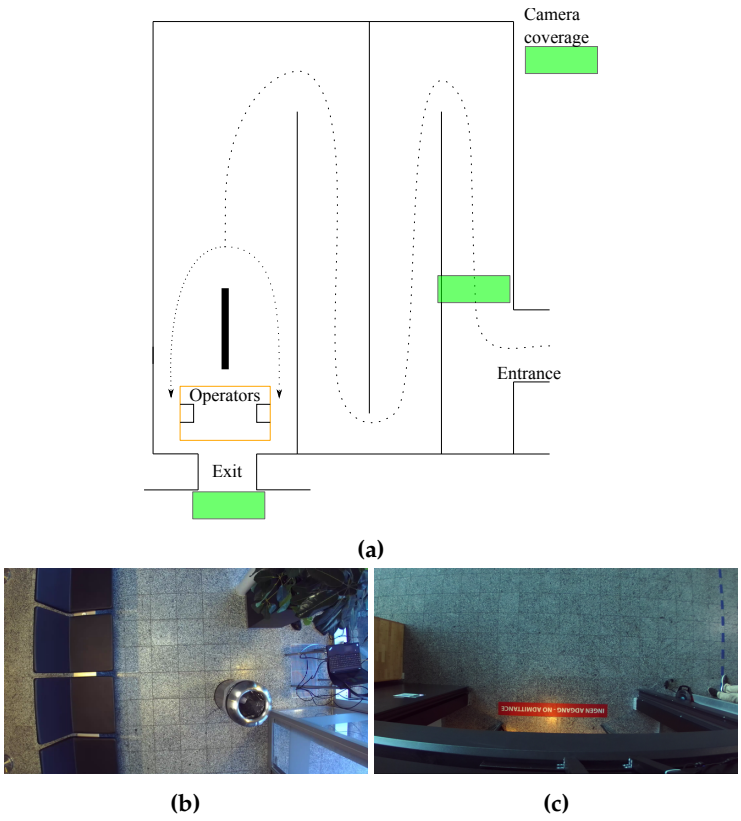


Fig. 3.2: (a) Maze to be followed by the persons in the queue, and camera coverage at entrance and exit. (b) Left view of the entrance camera and (c) left view of the exit camera.

2 Data Collection and Annotation

Similarly to chapter II.2, the ZED explorer was used to record the data. But differently, video was recorded in 2k resolution compared to FHD in chapter II.2 and the cameras were calibrated accordingly, as described in section II.1.2. We reduced the exposure time as much as possible to eliminate motion blur and make colors more distinguishable.

To make annotation of this dataset faster compared to OPR, a new video annotation tool has been programmed, which automatically annotates images based on detections. A screen shot of the tool is shown in Figure 3.3. The tool uses simple background subtraction to remove the floor and most static objects in the scene, which makes it easier to detect the persons. Since the camera is placed overhead pointing downwards, an obvious choice is to use the depth maps to determine how much background should be removed.

This is done simply by setting a *clipping distance* threshold, using the distances in the depth map, and remove anything above this threshold. Detections are then based on simple BLOB detection and bounding boxes are drawn to cover the blobs. The clipping distance threshold is set based on visual inspection of detections, to cover most of the persons within the bounding boxes, while removing most static objects. To avoid small bounding boxes around persons, effectively removing much of the persons, false detections have to be allowed, shown as the white box in Figure 3.3. To eliminate these, filter zones can be added, in which blobs are ignored, shown as green boxes. These can also be added to non-relevant areas where persons cannot appear anyway. The tool was used to automatically provide initial annotations that were afterwards manually refined by editing the bounding box sizes to fully cover the persons.

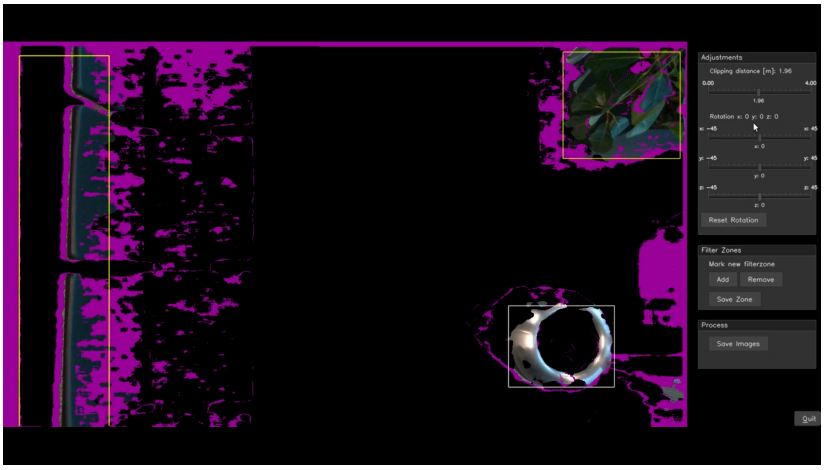


Fig. 3.3: Video annotation tool for QPR dataset. Example from the entrance. *Clipping distance* is set based on a trade-off between false detections and proper detections of persons.

3 Data Statistics

The dataset statistics of QPR are shown in Table 3.1. A total of 116 persons have been annotated, resulting in 11,616 bounding boxes, which are split more evenly between entrance and exit compared to OPR. On average, there are 100 bounding boxes per person.

The distribution of bounding boxes across persons is, furthermore, shown in Figure 3.4 (a) and (b). It is clear that as the size of the queue increases, more persons are present at the entrance for a longer period of time. Nonetheless, for the majority of the persons, the number of bounding boxes lie in the range

	Entrance	Exit	Total
Persons	116	116	116
Bounding boxes	7529	4087	11616
Avg. bounding boxes	69.91	35.23	100.14

Table 3.1: Overview of QPR dataset.

[50,100]. Finally, using timestamps from the two ZED cameras, Figure 3.4 (c) shows the actual queue times of each person, starting from the entrance of the first person at time zero. The first few passengers have more or less identical queue times, however, from time 100, the queue time increases linearly. Interestingly, it also seems as if there are two parallel lines increasing linearly. This could be due to either a staff manually deciding on which persons should be checked by each operator or due to cases where multiple persons that travel together are checked simultaneously by the same operator.

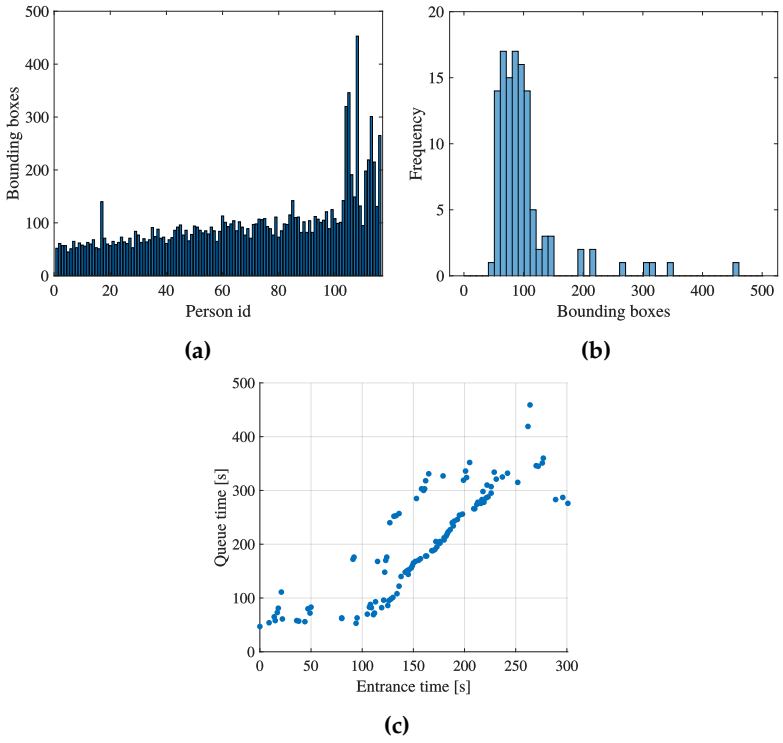


Fig. 3.4: (a) Number of bounding boxes per person, (b) frequency in number of bounding boxes in bins of 10, and (c) ground truth queue time based on time of entrance starting at zero for the first person. *Images are also used in [14].*

Chapter 4

Summary

Two datasets have been collected that consider re-id in a queue. First, a dataset, OPR, was collected in a university canteen, where persons would stand in queue upon entering the canteen and exit the queue when leaving the canteen. The dataset was collected from a single camera, which was placed overhead, during lunch time when the number of persons in the canteen would peak. The ZED Explorer tool was used to record the data, while the VATIC tool was used to annotate, which resulted in 64 annotated persons and in total 44,626 bounding boxes. This dataset has been used to evaluate novel multimodal features.

The second dataset, QPR, was collected from an immigration area in an airport, where the queue would be similar to those also seen in other areas of the airport. In this case, entrance and exit points were at different locations, hence, two non-overlapping cameras were used to collect data. Different from OPR, persons were constrained to follow a maze from entrance to exit, while the number of persons is higher. This dataset was annotated using a custom tool, which performs detections based on background subtraction and BLOB detection, that were afterwards manually refined. This resulted in a dataset, which contains 11,616 bounding boxes of 116 persons. Besides RGB and depth data, timestamps were saved, thus, this dataset has been used to both evaluate multimodal features and queue time measurements using vision-based re-id.

For both datasets, ZED cameras were used to record data, due to their wide FoV and high depth resolution. In case of OPR, the camera was calibrated and data were collected in FHD resolution, while in case of QPR, both cameras were calibrated and data were collected in 2k resolution. Depth maps are calculated using standard OpenCV function SGBM, and a JET color map is applied to transform the depth maps to images that can be used to train deep models.

References

[1] Asus. (2012) Xtion pro live. https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/overview/. Asus. Accessed: November 28, 2019.

[2] Orbbec. (2019) Astra series. https://orbbec3d.com/product-astra-pro/. Orbbec. Accessed: December 4, 2019.

[3] Microsoft. (2014) Kinect for windows. https://developer.microsoft.com/en-us/windows/kinect. Microsoft. Accessed: December 4, 2019.

[4] M. Azure. (2019) Azure kinect dk documentation. https://docs.microsoft.com/en-us/azure/kinect-dk/. Microsoft. Accessed: December 4, 2019.

[5] I. RealSense. (2018) Intel realsense depth camera d415. https://www.intelrealsense.com/depth-camera-d415/. Intel. Accessed: December 4, 2019.

[6] ——. (2018) Intel realsense depth camera d435. https://www.intelrealsense.com/depth-camera-d435/. Intel. Accessed: November 27, 2019.

[7] Stereolabs. (2017) Zed - depth sensing and camera tracking. https://www.stereolabs.com/zed/. Stereolabs. Accessed: November 28, 2019.

[8] OpenCV. (2019) Opencv 4.1.2. https://opencv.org/. OpenCV. Accessed: December 4, 2019.

[9] P. Greisen, S. Heinzle, M. Gross, and A. P. Burg, "An fpga-based processing pipeline for high-definition stereo video," *EURASIP Journal on Image and Video Processing*, vol. 2011, no. 1, p. 18, 2011.

[10] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[11] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, 2013.

[12] Topbulb. (2018) What is color rendering index (cri)? https://www.topbulb.com/color-rendering-index. Topbulb. Accessed: December 4, 2019.

[13] LITE. (2019) Panel light eco 60. https://lite-led.dk/produkt/panel-light-eco-60-40w-(4000k-white)-2?ChooseVariant=1. LITE A/S. Accessed: December 4, 2019.

[14] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "One-to-one person re-identification in a queue," Aalborg University, Tech. Rep., 2019.

Part III

Feature Extraction

Paper A

Multimodal Neural Networks for Overhead Person Re-identification

Aske R. Lejbølle, Kamal Nasrollahi, Benjamin Krogh, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 2017 International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 25–34, 2017.

Abstract

Person re-identification is a topic which has potential to be used for applications within forensics, flow analysis and queue monitoring. It is the process of matching persons across two or more camera views, most often by extracting colour and texture based hand-crafted features, to identify similar persons. Because of challenges regarding changes in lighting between views, occlusion or even privacy issues, more focus has turned to overhead and depth based camera solutions. Therefore, we have developed a system, based on a Convolutional Neural Network (CNN) which is trained using both depth and RGB modalities to provide a fused feature. By training on a locally collected dataset, we achieve a rank-1 accuracy of 74.69%, increased by 16.00% compared to using a single modality. Furthermore, tests on two similar publicly available benchmark datasets of TVPR and DPI-T show accuracies of 77.66% and 90.36%, respectively, outperforming state-of-the-art results by 3.60% and 5.20%, respectively.

1 Introduction

Person re-identification (re-id) i.e. identifications of persons across two or more cameras, is a topic with increasing interest due to potential usage in forensics, analysis of pedestrian flow in urban areas or monitoring of queue times in, for example, an airport. Meanwhile, it is also a topic still in research due to challenges that include changes in lighting, view and pose between camera views. To cope with these challenges, focus often lies in extracting robust hand-crafted feature descriptors from each person that are matched between views. For this purpose, soft biometrics are considered, such as colour and texture of the clothing, either represented as histograms [1] or transformed to sparse descriptors [2]. To further improve accuracy of correct matches, supervised learning algorithms are applied that learn to separate similar feature pairs from dissimilar ones [3, 4]. More recently, deep learning has drawn increasing interest from the research community with Convolution Neural Networks (CNN) outperforming hand-crafted feature descriptors, as they are able to learn more expressive features [5, 6].

Besides aforementioned challenges, privacy preservation is often related to person re-id as a potentially large amount of data needs to be stored. Other than representing images as feature descriptors, camera placement can be considered as a means of privacy preservation. Most current benchmark datasets within re-id consider a frontal view [7, 8] while only few consider an overhead view which has the advantage of reducing privacy issues and avoid occlusions between persons or objects and persons in the scene [9, 10]. Furthermore, other modalities that are more anonymous can be used, for example depth, from which information is captured using either passive stereo,

i.e. a stereo camera or active, for example, a Microsoft Kinect. From depth information, the height and width of the person can be extracted along with different body ratios [11]. Instead of relying on a single modality, combining (fusing) different modalities have shown to improve performance in related applications such as object recognition [12] and object segmentation [13]. Such fusing can be done either at feature level (feature fusion), for example, by concatenation of respective feature descriptors or at decision/score level (late fusion) by fusing the output decisions/scores from different modalities [14].

To consider challenges regarding changes across views and the advantages of fusing different modalities, we propose a novel framework for applying colour and depth (RGB-D) based re-id to images, captured with an overhead view. More specifically, we take advantage of the recent advances within deep learning and train a CNN using information from both RGB and depth modalities to improve accuracy compared to using either modality independently. To that extend, we collect a novel RGB-D based dataset in an uncontrolled environment from a stereo camera placed overhead to avoid occlusions and, at the same time, preserve privacy by not recording faces. Our dataset is collected to resemble real-life situations by having multiple persons within view, while current overhead datasets only consider a single person within view at a time. In summary, the main contributions of our work include:

- We train a CNN using RGB and depth modalities information and show that fusion of these improves accuracy.
- We collect and annotate a novel RGB-D and overhead based dataset which can be used to both evaluate re-id accuracy but also multi-target detection and tracking algorithms in RGB and depth domain.

2 Related Work

While re-id using hand-crafted colour and texture features or CNN's are widely studied, overhead re-id is rarely considered. In addition, only a limited number of articles suggest depth modality for this purpose.

Overhead re-id As most current re-id datasets are collected in outdoor scenes, a frontal view is typically considered. A few systems have been proposed for evaluating datasets with an overhead view [15, 16]. [16] proposes feature extraction using a Histogram of Oriented Gradients (HOG) algorithm combined with a linear Support Vector Machine (SVM) for classification while [15] extracts features based on the colour and texture of the hair. While both datasets are recorded in an indoor environment, they only

extract colour information.

Overhead RGB-D-based re-id More RGB-D based datasets for re-id are currently being proposed. While the first considered a frontal view [17], the most recent consider an overhead view [9, 10]. [9] collected a dataset in a hallway and applies a combined CNN and Long-Short-Term-Memory (LSTM) network using depth based image sequences to learn spatio-temporal representations of each person. Meanwhile, [10] extracts seven different depth features and two colour features that are feature fused by concatenation. While the former extracts only depth information, the latter considers only hand-crafted features from both modalities.

Multi-modal CNN While the work of [9] to our knowledge is the only previous proposed neural network using depth information for re-id, multi-modal CNN's have been proposed for related applications [12, 13]. [12] trains a CNN for object recognition using both colour and depth images by fusing respective features in late layers of the network to consider both modalities during training. To that extend, [18] shows that feature fusion of colour and depth features in a CNN outperforms similar fusion scheme using other classification methods, such as SVM and Deep Belief Networks (DBN). Meanwhile, [13] proposes a multi-modal encoder-decoder network for semantic segmentation by fusing outputs from each layer in an RGB and depth based encoder, respectively, before passing the output through an RGB-D based decoder. In this case, fusing is applied as an element-wise summation. To our knowledge, no multi-modal neural networks have previously been proposed for re-id. Although, [6] proposes a fusing scheme similar to that of [12], but instead of fusing different modalities, complementary feature types are fused, i.e. CNN and hand-crafted features. To our knowledge, the system proposed in this paper, is the first to incorporate multiple modalities in a CNN to learn a multi-modal feature representation.

3 Methodology

As we desire to exploit both colour and depth information, along with the potential of CNN's, our aim is to use an architecture which jointly processes the two modalities, RGB and depth, simultaneously. For person re-id, such architecture has not previously been applied, although, in object recognition the work of [12] shows an increase in accuracy compared to using a single modality.

We apply an architecture similar to that of [12], having two CNN streams separately processing an input image while being fused in a later fully connected layer, as shown in Fig. A.1. The structure of each separate CNN

follows the AlexNet architecture (please see [19] for details) and consists of five convolution layers, the first, second and fifth followed by a max pooling and normalization layer. The outputs from the last convolution layers are followed by two fully connected layers, transforming the feature maps to sparse representations for RGB and depth, respectively. The feature representations ($fc7^{RGB}$ and $fc7^{D}$) are concatenated and used as input to a fully connected layer (fc8) which learns a joint RGB-D feature representations based on both colour and depth images. Finally, a softmax layer (fc9) calculates output probabilities for each class, defined as a person ID, which combined with a loss function is used to update the parameters of the network. We refer to our proposed system as RGB-D-CNN. At test time, the softmax layer is discarded and features are extracted from fc8.

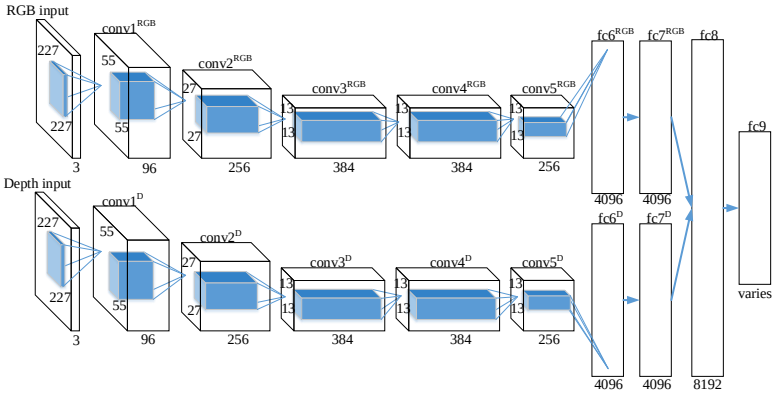


Fig. A.1: Overview of the RGB-D based CNN (RGB-D-CNN). Lower part processes a depth image, while the upper part processes a colour image, features from last fully connected layer of the streams are fused in a joint fully connected layer before classification.

Individual training Before training the RGB-D-CNN, CNN models are trained for RGB and depth, respectively. We refer to these models as RGB-CNN and D-CNN. Both follow similar structure as the upper/lower part of the joint CNN, with a softmax layer replacing fc8 and fc9. The model weights are initialized using a pre-trained model of the CaffeNet version [20] of AlexNet trained on the ImageNet dataset. Following the architecture of AlexNet, the input is an image of size 227×227, randomly cropped from an image of size 256×256, to make the network robust to changes in translation. Both colour and depth images are therefore resized accordingly before being processed by the network. In addition, the images are randomly flipped to increase the amount of training data. In case of depth images, [12] shows that applying a jet colourmap enhances the accuracy compared to encoding

the images using surface normals [21] or Horizontal disparity, Height and Angle (HHA) encoding [22]. This colour transformation maps each depth value to a colour in RGB colour space from blue(close) over green to red(far). This enables us to initialize the weights using the pre-trained CaffeNet model without additional preprocessing. We therefore perform similar step before training the depth model.

Given sets of parameters and datasets $(W^{RGB}, b^{RGB}, X^{RGB}, Y)$ and (W^{D}, b^{D}, X^{D}, Y) for RGB and depth, respectively, where W and b are the model weights and bias, while (X^{RGB}, X^{D}) are the set of RGB and depth images with corresponding labels Y, we train the models by minimizing a loss function, L, as given in Equation A.1:

$$\begin{aligned} &\min_{W,b} -\frac{1}{N}\sum_{i=1}^{N} L(W, b, x_i, y_i) \\ &L(W, b, x_i, y_i) = log(\hat{p}_i, y_i) \end{aligned} \tag{A.1}$$

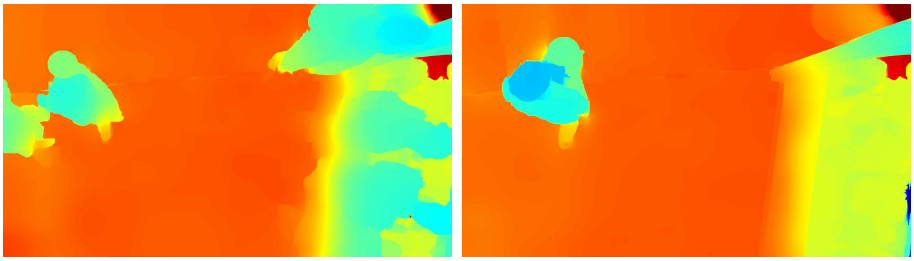
where W, b are the weights and bias of the model currently being trained, $X = \{x_1, .., x_N\}$ is the sample set and $\hat{p}_i$ is the output probability from the softmax layer of the i'th sample given the true label y_i.

Joint training After training RGB-CNN and D-CNN, the model parameters are used to initialize the two CNN streams in RGB-D-CNN. The softmax layers are replaced by a randomly initialized fully connected layer (fc8) and new softmax layer (fc9). By fusing outputs from both fc7^{RGB} and fc7^{D} in fc8, the parameters of the depth stream are updated depending on the input to the RGB stream and vice versa, while the weights and bias of fc8 are updated based on both inputs, resulting in a fused output. [6] shows how fusion of hand-crafted and CNN features in the late layers of the network affects parameter update of the CNN. Similar proof applies to this context.

4 Experimental Results

Datasets For evaluation we present a novel RGB-D based dataset collected from an overhead view. We refer to the dataset as Overhead Person Re-identification (OPR). The dataset is collected using a calibrated ZED stereo camera from Stereolabs [23], mainly due to its ability to record depth from a range 0.7m-20m covering both low and high ceilings. In addition, it captures video in resolutions up to 4416×1242 pixels which is much higher than RGB sensors in solutions such as the Microsoft Kinect. The camera is placed in the ceiling at a university canteen (uncontrolled environment) to capture a populated area. From this perspective, persons are captured when approaching (walking from top to bottom), and leaving (walking from bottom to top)

the canteen a few minutes later, enabling us to evaluate re-id performance. Data is collected on a single day during a two hour period around midday to capture video when the number of persons in the canteen is increasing and decreasing. As a result, cases of having a large number of persons and only a single person are recorded, examples of captured depth images in both cases are shown in Fig. A.2 (a) and (b), respectively.



(a) (b)

Fig. A.2: Examples of depth images containing (a): multiple persons and (b): containing a single person. Each person is captured when approaching (right side) and leaving (left side) the canteen.

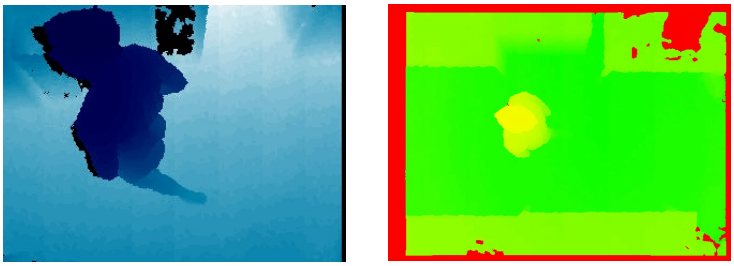
Disparity maps are computed using Semi-Global Block Matching (SGBM) as it has shown as a good compromise between accuracy and processing time [24], followed by filtering using a Weighted Least Square (WLS) kernel to eliminate noise and make the background more uniform, resulting in more precise depth information. Finally, we manually annotate bounding boxes around persons and use those for our system, the annotations enables us to further test detection, tracking and segmentation algorithms in future work. A total number of 78742 frames with 64 different persons have been annotated for re-id.

To our knowledge, only the datasets of [9] (DPI-T) and [10] (TVPR) have previously been proposed for RGB-D and overhead based re-id. Both are recorded in a hall with only a single person within view at all times. Examples of depth images from these datasets are shown in Fig. A.3. In addition to evaluating on our own dataset, we apply our system to those of [9] and [10] for comparison with their original results.

Evaluation protocols Depending on the dataset, different training and testing protocols are followed.

OPR Similar to most RGB-based datasets within re-id, we perform 10 random train and test splits, each set containing 32 persons. After training the CNN models, features from the test set are extracted from the last fully connected layer.

TVPR The training set consists of 100 persons walking from left to right while



(a) (b)

Fig. A.3: Examples of depth images from (a): DPI-T and (b): TVPR.

the test set consists of same persons walking from right to left. During test, features from the test set are compared with those from the training set. Although, due to issues at test time regarding one of the video sequences, only 94 persons were considered for training and testing.

DPI-T 12 persons appear in five different sets of clothing in both the training and test, while the number of recordings in each set differs. A total of 213 sequences are used for training while the test set consists of 249 sequences which are all classified by comparing with those of the training set.

When training RGB-CNN and D-CNN, a batch size of 128 is used while a size of 64 is used in case of RGB-D-CNN. Network parameters are updated using Stochastic Gradient Descent (SGD) with momentum is to avoid getting stuck in a local minimum. Hyper parameters are set accordingly to [12] with a momentum of 0.9 and base learning rate of 0.01 which is reduced by multiplying with 0.97 for each epoch. At each epoch, the training set is randomly shuffled for faster convergence [25]. We present our results by calculating the *rank-1* to *rank-k* accuracies based on feature matching where *rank-i* indicates a cumulative percentage of persons having their true match within the i most similar with k indicating the total number of persons. For OPR, the average accuracies over all train/test splits are calculated. Matches are calculated using Euclidean distance between extracted features following a multi-shot approach, i.e. features from all images of each person/sequence are extracted and either maximized or averaged, indicated by subscripts max and avg.

Fig. A.4 (a) shows the resulting Cumulative Matching Characteristic (CMC) curves for applying RGB-CNN, D-CNN and RGB-D-CNN to OPR. It is clear that fusing of RGB and depth modalities clearly increases accuracy compared to using a single modality. The best result is achieved by RGB-D-CNN$_{avg}$, increasing accuracy by 16.00% compared to RGB-CNN$_{avg}$. Furthermore, Fig. A.4 (b) and (c) show the results of our system applied to DPI-T and TVPR, respectively. In case of DPI-T, RGB-D-CNN$_{avg}$ still outperforms RGB-CNN and D-CNN with an increase of 3.61% compared to RGB-CNN$_{avg}$. Finally for TVPR, RGB-CNN provide better results compared to RGB-D-CNN. A reason

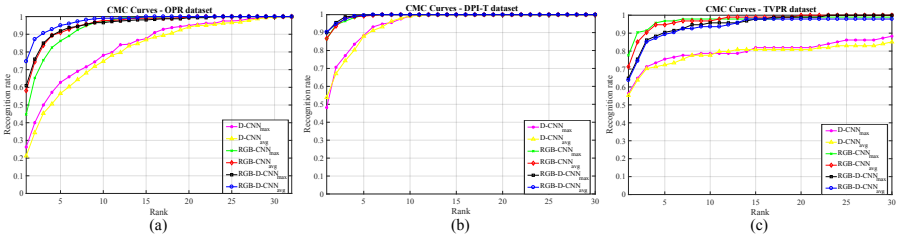


Fig. A.4: Results on (a) OPR (p=32), (b) DPI-T (p=249) and (c) TVPR (p=94) for RGB-CNN, D-CNN and RGB-D-CNN, respectively, using maximized (*max*) and averaged (*avg*) features.

for this could be the quality of depth information (see Fig. A.3 (b)) negatively affecting the training of RGB-D-CNN in combination with corresponding colour images. Even though, D-CNN results are slightly worse in case of DPI-T, the level of detail in depth images are higher (see A.3 (a)) causing the modality to better complement RGB. The quality of depth information therefore seems important when training an RGB-D CNN. Looking at results across all datasets, averaged features mostly provides the highest accuracies, although, in case of depth features, feature maximization seems better. This could be due to encoding of features as colourized images combined with an overhead view from which the height of each person, and thereby the colour gradient, is important. By averaging features, this information more easily gets lost if the representation changes between images.

Tables A.1 summarizes our results on TVPR and DPI-T, compared to their original results. As [9] only provides a rank-1 accuracy while [10] only provides CMC curves, only the rank-1 accuracy is considered. For [10], rank-1 is estimated from the CMC curves. *Ours* refers to the best results achieved by our system (RGB-D-CNN$_{avg}$ in case of DPI-T and RGB-CNN$_{avg}$ in case of TVPR). In both cases we outperform original results, for DPI-T by 34.76% by also using RGB. From Fig. C.3 (b), it is worth noting that our D-CNN alone achieves almost similar accuracies as [9] who also adds an LSTM layer on top of a similar CNN.
Even though, six persons are missing for the tests on TVPR, our system shows potential to be improved further. For RGB alone, our system outperforms that of [10] by ≈5.16%.

Processing time We evaluate processing time for stereo and feature matching on OPR to discuss on the potential of using passive stereo for re-id applications. 20 matching iteration are run using an Intel i7-6700HQ CPU @ 2.60GHz and 16GB of RAM and average timings are provided. Stereo matching matching is performed on images of size 960×540.

While feature matching only takes 4.0e10^{-5}s, SGBM and WLS are more processing intensive taking 0.136s and 0.103s, respectively. Nonetheless, ≈4

	Rank-1 accuracy [%]	
Method	*DPI-T*	*TVPR*
4D RAM [9]	55.60	–
TVDH [10]	–	72.50
Ours	**90.36**	**77.66**

Table A.1: Comparison of our RGB-D-CNN to original results on DPI-T and TVPR datasets.

FPS is achieved using the CPU. For real-time applications, GPU implementations of SGBM and WLS algorithms could be used speed up the process. No such implementations are available at the moment.

5 Conclusion

In this paper, we have presented an RGB-D based CNN applied to person re-identification. Two CNN models are trained using colour and depth images, respectively, captured from an overhead view and resulting trained parameters are used to initialize a joint RGB-D-CNN model trained using both modalities. To test the system, we collected a novel RGB-D and overhead based dataset which is annotated for evaluation on both re-id accuracy, but also detection and tracking algorithms. By applying our system to our novel and two previously proposed datasets, we have shown that the combination of RGB and depth modalities increase accuracy by 16.0% and 3.6% on our OPR dataset and DPI-T, respectively. In case of TVPR, RGB modality alone achieved higher accuracy than combining modalities due to the quality of depth information. This indicates an importance to capture detailed depth information to proper complement the RGB modality. In addition, our system shows an FPS of 4 using a CPU, with potential of being increased if processing intensive algorithms such as SGBM and WLS are implemented on a GPU. For future work, the system should be evaluated on bounding boxes extracted automatically from a person detector. To increase detection performance, depth information could also be used for this purpose. Furthermore, our proposed system could be extended with an LSTM to handle video rather than averaging or maximizing features extracted from a sequence of images. This would allow for temporal information to be captured as well. Finally, more recently developed neural networks could replace the AlexNet architecture to increase performance and decrease processing time.

Acknowledgement

The work carried out in this paper is supported by Innovation Fund Denmark under Grant 5189-00222B.

References

[1] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[2] S. Li, M. Shao, and Y. Fu, "Cross-view projective dictionary learning for person re-identification," in *Proc. AAAI*. AAAI Press, 2015, pp. 2155–2161.

[3] Y.-C. Chen, W.-S. Zheng, J.-H. Lai, and P. Yuen, "An asymmetric distance model for cross-view feature mapping in person re-identification," *IEEE Transactions on Circuits and Systems*, vol. 27, no. 8, pp. 1661–1675, 2016.

[4] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. CVPR*, 2016, pp. 1239–1248.

[5] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. CVPR*, 2015, pp. 3908–3916.

[6] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[7] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[8] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[9] A. Haque, A. Alahi, and L. Fei-Fei, "Recurrent attention models for depth-based person identification," in *Proc. CVPR*, 2016, pp. 1229–1238.

[10] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, "Person re-identification dataset with rgb-d camera in a top-view configuration," in *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 1–11.

[11] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino, "Re-identification with rgb-d sensors," in *Proc. ECCV*. Springer, 2012, pp. 433–442.

[12] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Proc. IROS*, 2015, pp. 681–687.

[13] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Proc. ACCV*. Springer, 2016, pp. 213–228.

[14] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[15] H. Aradhye, M. Fischler, R. Bolles, and G. Myers, "Headprint-based human recognition," in *Advances in Biometrics: Sensors, Algorithms and Systems*. Springer London, 2008, ch. 15, pp. 287–306.

[16] I. Ahmed and J. N. Carter, "A robust person detector for overhead views," in *Proc. ICPR*. IEEE, 2012, pp. 1483–1486.

[17] M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti, "3d reconstruction of freely moving persons for re-identification with a depth sensor," in *Proc. ICRA*. IEEE, 2014, pp. 4512–4519.

[18] J. Sanchez-Riera, K.-L. Hua, Y.-S. Hsiao, T. Lim, S. C. Hidayati, and W.-H. Cheng, "A comparative study of data fusion for rgb-d based visual recognition," *Pattern Recognition Letters*, vol. 73, pp. 1–6, 2016.

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

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[21] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Proc. Experimental Robotics*. Springer, 2013, pp. 387–402.

[22] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Proc. ECCV*. Springer, 2014, pp. 345–360.

[23] Stereolabs. (2017) Zed - depth sensing and camera tracking. https://www.stereolabs.com/zed/. Stereolabs. Accessed: November 28, 2019.

[24] R. Kalarot, J. Morris, D. Berry, and J. Dunning, "Analysis of real-time stereo vision algorithms on gpu," in *Proc. IVCNZ*, 2011.

[25] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

Paper B

Attention in Multimodal Neural Networks for Person Re-identification

Aske R. Lejbølle, Benjamin Krogh, Kamal Nasrollahi, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 179–187, 2018.

The layout has been revised.

Abstract

In spite of increasing interest from the research community, person re-identification remains an unsolved problem. Correctly deciding on a true match by comparing images of a person, captured by several cameras, requires extraction of discriminative features to counter challenges such as changes in lighting, viewpoint and occlusion. Besides devising novel feature descriptors, the setup can be changed to capture persons from an overhead viewpoint rather than a horizontal. Furthermore, additional modalities can be considered that are not affected by similar environmental changes as RGB images. In this work, we present a Multimodal ATtention network (MAT) based on RGB and depth modalities. We combine a Convolution Neural Network with an attention module to extract local and discriminative features that are fused with globally extracted features. Attention is based on correlation between the two modalities and we finally also fuse RGB and depth features to generate a joint multilevel RGB-D feature. Experiments conducted on three datasets captured from an overhead view show the importance of attention, increasing accuracies by 3.43%, 2.01% and 2.13% on OPR, DPI-T and TVPR, respectively.

1 Introduction

Person re-identification (re-id) is the task of matching person descriptors extracted from images captured across, typically, non-overlapping cameras and persists as a hot topic within the research community [1]. This is not only due to major challenges, including changes in lighting, viewpoint and occlusion between cameras but also the potential usage within applications such as forensics or long-term tracking of pedestrians [2, 3].

A person re-id system, typically, consists of tracking, features extraction and feature matching using simple distance metrics, for example, Euclidean distance or more sophisticated ones such as Keep It Simple and Straightforward MEtric (KISSME) based on Bayesian theory and Mahalanobis distance [4]. Variations such as Cross-view Quadratic Discriminant Analysis (XQDA) [5] additionally considers subspace learning while Support Vector Machines (SVM) [6] aims at maximizing distance between features of non-similar pairs. For a comprehensive overview of metrics applied in person re-id, please see [7]. Most often, researchers focus on either feature extraction or matching using supervised learning, although, following recent developments of deep learning, Convolution Neural Networks (CNN) have been proposed also in the case of person re-id [8–13]. These networks are able to learn both discriminative features and a classifier simultaneously by training in an end-to-end fashion.

Due to more focus on CNN in re-id, more data has become a necessity to properly train the networks. As a result, larger datasets in recent years have

emerged [14–16], not only allowing proper use of CNN but also increasing the realism of re-id evaluation. Common for these datasets is the viewpoint which is mostly horizontal, allowing occlusions between persons or persons and objects. Another option is to place the camera overhead, resulting in a vertical viewpoint, an option only considered by few [17–19]. This both has the potential of decreasing the probability of occlusions and improve privacy preservation. Examples of the two different viewpoints are shown in Figure B.1.

(a) (b)

Fig. B.1: Examples of images captured from (a): an overhead viewpoint [17] and (b): a horizontal viewpoint [20].

By changing the viewpoint, less color and texture information might be available and it is therefore crucial to extract features that represent the most important parts of a persons appearance. One way is to learn part-specific CNN models by splitting the image into local regions and feed those to separate CNN streams [10, 12]. Even though, these models learn local feature responses, they still consider regions that are not relevant to the feature descriptor, decreasing invariance to lighting, background clutter, etc.

Another way is to apply an *attention mechanism*, originally introduced and applied in Neural Machine Translation problems (NMT) [21], which can be used to consider only certain local parts of an image. Within computer vision, this method has been applied with great success to both image captioning [22], action recognition [23] and, more recently, person re-id [17, 24, 25].

Attention works by calculating a set of positive weights defined as a 2D attention map. Attention maps are then used to summarize features extracted from a CNN. Two types of attention are often considered, soft attention where attention weights are calculated based on a differentiable deterministic mechanism which can easily be trained along with the rest a neural network, and hard attention where weights are calculated by a stochastic process.

Besides capturing local information, additional modalities can be consid-

ered to extract different heuristics. Based on extracting features from images captured in an overhead view, it makes sense to include depth information as an additional modality. To that end, previous work on multimodal person re-id has shown RGB and depth based features to complement each other well [18, 19, 26].

In this paper we apply soft attention to person re-id, considering images captured from an overhead view. Instead of only applying attention using color or depth information, we consider a multimodal approach by calculating attention weights based on fusing RGB and depth features, both extracted using pre-trained CNN. As a result, attended regions in the RGB image are based on the representation in depth domain which better captures information around regions with significant change in depth. Vice versa, attended regions in the depth domain are based on the RGB representation to better capture depth information in areas with discriminative color information. To extract features from different discriminative regions, we learn attention maps at multiple layers of the CNN and fuse locally summarized features. Additionally, local features are fused with global feature descriptors to capture information at different abstraction levels as previously proposed with success [11, 27].

Finally, we also learn a joint feature representation by fusing RGB and depth features in the late layers of the network to produce a multilevel RGB-D based feature descriptor and train the entire network end-to-end. To summarize, our contributions include:

- We implement soft attention in a multimodal CNN by fusing RGB and depth features.
- We analyze the importance of attention in a multimodal context by visualizing calculated attention maps in different scenarios.

The rest of the paper is structured as follows. Related work is presented in Section B.2 followed by a description of the proposed methodology in Section B.3. Experimental results are presented in Section B.4, including an impact analysis of applying attention. Finally, the paper is concluded in Section B.5.

2 Related Work

Ever since the first significant results in object recognition [28], CNN have been proposed in person re-id [8, 14]. While these focus on globally extracted features, more recent proposals are based on part-based learning to capture more local information [9, 10, 13]. Ustinova *et al.* [10] propose a Bilinear-CNN by splitting the body into three parts and train part specific CNN that are summarized by bilinear combination of features. Finally, features from

the three parts are fused in a fully connected layer. Part specific CNN are also proposed by Cheng *et al.* [9] who split the body into four parts and learn both part specific and global features that are fused in the late layers of the CNN. A different approach is followed by Zhao *et al.* [13] who apply a Regional Proposal Network (RPN) to locate 14 human body joints and extract seven body sub-regions. A CNN is applied to each sub-region to learn part specific features that are afterwards fused in a four layered feature fusion network (FFN). Part localization is also proposed by Li *et al.* [29], but instead of localizing the joints, they apply a Spatial Transformer Network (STN) to localize head-shoulder, upper-body and lower-body regions. Once again, part specific features are learned and later fused with globally extracted features. Common for aforementioned methods is the requirements of a horizontal viewpoint in order to either have a properly division of body parts or localize the joints. In case of an overhead view, this is not possible.

Soft attention in CNN can be related to saliency learning using handcrafted features which also aims at locating discriminative regions. Little work has been done within this area, most notable are the works of Zhao *et al.* [30, 31]. In [31] they propose salience learning by matching patches within a constrained window between images of persons captured by two different cameras. For each patch, a salient score is calculated using either K-Nearest Neighbors or One-class SVM. Meanwhile, in [30] they propose learning discriminative mid-level filters by clustering image patches with coherent appearance and apply SVM. These filters are then used to calculate filter responses of input images prior to feature matching.

Attention has been previously proposed only a few times within person re-id [17, 24, 25]. Liu *et al.* [24] propose a Comparative Attention Network (CAN) which is trained end-to-end by producing and comparing attended regions of positive and negative image pairs, i.e., images of similar and non-similar persons. By combining a CNN with a Long Short-Term Memory (LSTM) network, attention maps are produced at different time steps to capture different local regions by using the same encoded image as input at each time step. The work of Zhao *et al.* [25] is also motivated by attention which is used to model a part-aligned human representation by learning attention weights through end-to-end training using a triplet loss function. Finally Haque *et al.* [17] propose a depth-based recurrent visual attention network by combining a CNN with an LSTM to learn spatiotemporal features. By adding a localization network, discriminative features are extracted from glimses, i.e., a minor region in the input. The localization network is trained using reinforcement learning to focus on discriminative regions. While [24, 25] apply attention in the RGB domain, [17] apply attention in depth domain. This work, to our knowledge, is the first to apply attention in a multimodal context.

Multimodal fusion of RGB and depth information is rarely considered in

person re-id [19, 32, 33]. Liciotti *et al.* [19] propose a combination of hand-crafted RGB and depth features to capture both color, texture and anthropometric information. RGB-D based hand-crafted features are also proposed by Wu *et al.* [33] who extract a rotation invariant Eigen-depth feature and fuse it with low-level color and texture features [5]. Only two previous proposals fuse RGB and depth features using a CNN [18, 26]. Karianakis *et al.* [26] learn spatiotemporal features from a combined CNN and LSTM. Considering the small sample size issue, they add hard attention to incorporate regularization. Finally, Lejbølle *et al.* [18] propose a multimodal CNN which jointly learns a multimodal feature descriptor based on individually trained RGB and depth CNN. Common in aforementioned work is fusion of features which is simply done by concatenation which does not capture the correlation between features from different modalities. In this work, we use correlation between depth and RGB features to extract local information from the input images and, additionally, exploit the advantage of multimodal feature fusion by learning a joint descriptor based on RGB and depth.

3 Methodology

An overview of the proposed network is shown in Figure B.2. RGB and depth images I_{RGB} and I_D are encoded using an RGB based encoder f_{RGB} and depth encoder f_D, respectively, represented by CNN. The outputs from the last convolution layer are embedded in fully connected layers and used as input to the attention model f_{att}. The attention model multiplies features to capture correspondence between modalities, following the idea of multiplicative interaction [34]. Attention weights, α^l, are afterwards calculated separately for the l'th layer of the RGB and depth encoders, and used to summarize feature maps X^l_{RGB} and X^l_D. The summarized features are fused with globally extracted feature descriptors and the two modality based features are fused to learn a joint feature representation. Finally, a classification module f_c is added for prediction when training the network. In the rest of the paper, we refer to our proposed network as Multimodal ATtention network (MAT).

3.1 Visual Encoder

The input to the MAT is an RGB image I_{RGB} and a corresponding depth image I_D that are separately processed by modality based encoders f_{RGB} and f_D given by,

$$\begin{aligned} X^5_{RGB} &= f_{RGB}(I_{RGB}, \theta_{RGB}) \\ X^5_D &= f_D(I_D, \theta_D), \end{aligned} \tag{B.1}$$

where $X^5_{RGB} \in R^{N\times N\times K}$ and $X^5_D \in R^{N\times N\times K}$ are the outputs from the

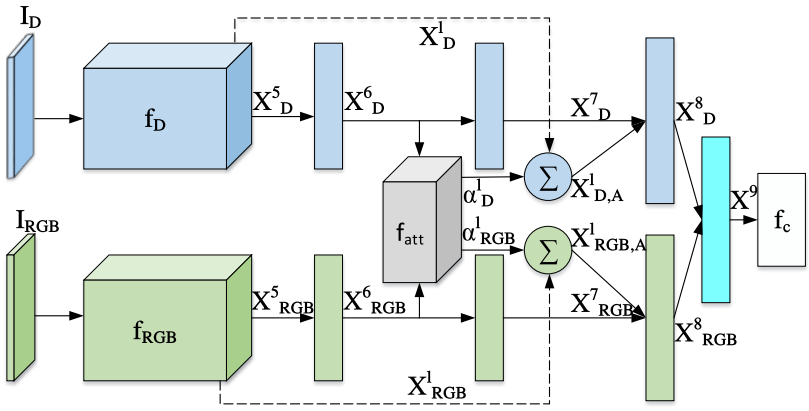


Fig. B.2: Overview of the Multimodal ATtention network (MAT). An RGB and depth image is encoded by an RGB based encoder, shown by the green stream, and depth based encoder, shown by the blue stream, respectively. Outputs from the last convolution layer are embedded and applied to the attention module which calculates attention maps for each modality. Feature maps from the encoders are summarized using the attention maps and fused with global feature representations at each modality. Finally, features from the two modalities are fused to a multilevel RGB-D based feature descriptor and used for prediction.

fifth and last convolution layer denoted by the superscript *5*, θ_{RGB} and θ_D are the encoder weights while K represents the number of feature maps of size $N \times N$.

The encoders follow the Caffe variation (CaffeNet) [35] of the AlexNet CNN [28] for better comparison with the related method of [18] which does not consider attention. The CaffeNet consists of five convolution layers, the first and second followed by local response normalization and max pooling. Max pooling is also added after the fifth convolution layer and followed by three fully connected layers, the last one used to calcuate an output score for each class normalized by a softmax function. Rectified Linear Units (ReLU) are used as nonlinear activation while dropout with a probability of 0.5 is added between fully connected layers to increase network generalization [36].

Followeing the baseline architecture of [18], the encoders take as input images of size 227×227 and output feature descriptors $X^5_{RGB} \in R^{13\times13\times256}$ and $X^5_D \in R^{13\times13\times256}$. Two fully connected layer afterwards embed features to sparse feature descriptors $X^7_{RGB} \in R^{4096}$ and $X^7_D \in R^{4096}$, representing global information. Different from [18], we do not fuse X^7_{RGB} and X^7_D to a joint RGB-D feature, but first fuse each modality based feature with locally extracted features from the attention model.

3.2 Attention Model

The attention model f_{att} is based on using depth information to calculate attention weights for the RGB features and vice versa. In this subsection, we outline the calculation of RGB attention weights, similar calculations are defined in case of depth by simply exchanging subscripts *RGB* and *D*.

As input to the attention model, we use features $X^6_{RGB} \in R^{4096}$ and $X^6_D \in R^{4096}$ extracted from the first fully connected layer. The attention weights to extract local features from the output of any given layer of f_{RGB} are then calculated as,

$$\begin{aligned} e^l &= f_{att}(X^6_{RGB}, X^6_D, \theta_\alpha), \quad e^l \in R^{N^2} \\ \alpha^l_i &= \frac{\exp(e^l_i)}{\sum_i e^l_i}, \qquad\qquad\qquad \alpha^l \in R^{N^2}, \end{aligned} \tag{B.2}$$

where e^l is a vector of unnormalized attention weights of size N^2 and θ_α represents the attention model parameters. To calculate a weighted average of features, attention weights are normalized using a softmax function, resulting in α^l, as originally proposed [21]. Thus, given a feature map of, e.g., size 13×13, we calculate 169 normalized attention weights.

The attention model implements multiplicative interaction to learn relations between RGB and depth features, and calculation of attention weights can therefore also be written as,

$$e^l = W^l_{att}(X^6_{RGB} \odot X^6_D) + b^l_{att}, \tag{B.3}$$

where $\odot$ represents an element-wise multiplication while $W^l_{att} \in R^{M \times N}$ and $b^l_{att} \in R^N$ are the weights and bias of the attention model, respectively, and M is the number of hidden units in X^6_{RGB}.

The normalized attention weights calculated in Equation B.2 are then used to calculate the weighted average of features from the l'th layer of f_{RGB} as,

$$X^l_{RGB,A} = (\hat{X}^l_{RGB})^T \alpha^l, \quad X^l_{RGB,A} \in R^K, \tag{B.4}$$

where $\hat{X}^l_{RGB} \in R^{N^2 \times K}$ is the flattened output from layer l and $X^l_{RGB,A}$ is a feature descriptor containing local information from the input RGB image dependent on features from the depth image. Since the attention maps are used to summarize features across all feature maps, only local regions of interest are considered. In our experiments presented Section C.4, we calculate attention maps for the fourth and fifth convolution layers of f_{RGB} and f_D to capture different local information, resulting in calculations of, in total, four attention maps. We observe that consideration of additional low-level information from earlier convolution layers does not improve accuracy. Given the outputs $X^4_{RGB} \in R^{13 \times 13 \times 384}$ and $X^5_{RGB} \in R^{13 \times 13 \times 256}$, we thereby summarize

features using attention maps $\alpha^4_{RGB} \in R^{169}$ and $\alpha^5_{RGB} \in R^{169}$ resulting in attention based features $X^4_{RGB,A} \in R^{384}$ and $X^5_{RGB,A} \in R^{256}$.

The attention based features are afterwards fused with X^7_{RGB} by adding a new fully connected layer, to form a modality based multilevel feature $X^8_{RGB} \in R^{4096}$. Finally, multilevel RGB and depth features are fused by a second new fully connected layer resulting in a multimodal feature descriptor X^9_{RGBD} used for prediction.

Prediction is implemented by calculating a probability score of each class given X^9_{RGBD}. A softmax layer is added to normalize scores and the entire network is trained end-to-end by minimizing the logistic loss function defined as,

$$\begin{aligned} &\min_{\theta_{RGB},\theta_D,\theta_\alpha,\theta_c} -\frac{1}{J}\sum_{i=1}^{J} log(\hat{p}_i) \\ &\hat{p}_i = f_{MAT}(I_{RGB}, I_D; \theta_{RGB}, \theta_D, \theta_\alpha, \theta_c), \end{aligned} \tag{B.5}$$

where the loss is calculated over a mini-batch of size J and $\hat{p}_i$ represents the normalized score for the i'th image predicted by f_{MAT}.

4 Experiments

This section outlines the experimental results and analysis of the MAT. First, evaluated dataset and corresponding test protocols are described followed by details of training f_{RGB}, f_D and f_{MAT}. Finally, the results are presented with a comparison to state-of-the-art methods and the attention module is analyzed by a visualization of calculated attention maps.

4.1 Datasets and Protocols

When evaluating the MAT, we only consider datasets collected from an overhead viewpoint. Three RGB-D based datasets, to our knowledge, have been proposed for overhead person re-id, including the Depth-based Person Identification from Top (DPI-T) [17], Top View Person Re-identification (TVPR) [19] and Overhead Person Re-identification (OPR) [18].

DPI-T: Recorded in a hallway, this dataset consists of 12 persons, appearing in an average of 25 sequences in five different appearances, both in the training and test set. A total of 213 sequences are included in the training set, while 249 are used for testing. During test, all sequences from the test set are matched with those in the training set.

TVPR 23 videos are recorded in a hallway, including a total of 100 persons, each appearing twice. The training set consists of persons walking from left to right, while walking from right to left in the test set. At test time, sequences from the test set are compared with those of the training set. Due to

missing frames in one of the recorded videos at time of testing, 94 persons are considered in our evaluation. Different from [18] who consider full-frame images, we apply a You Only Look Once (YOLO) detector [37] optimized for person detection, to automatically extract the ROI around persons.

OPR This dataset, recorded in a university canteen, consists of 64 persons captured twice, when entering and leaving the canteen. Similar to protocols in widely used re-id datasets captured from a horizontal view, 10 random train/test splits are performed, each consisting of 32 persons in both training and test set. The final result is then calculated as an average of accuracies across all experiments.

4.2 Implementation details

Before training the MAT, f_{RGB} and f_D are fine-tuned by initializing a Caffe model, pre-trained on the ImageNet dataset. In case of f_D, we encode depth images by applying a JET colormap which has shown to outperform other encoding methods such as surface normals or Horizontal disparity, Height and Angle (HHA) [38]. In addition to also being faster, applying a color map allows us to initialize weights using a pre-trained ImageNet model since each depth value is mapped to a value in the RGB color space ranging from blue (close to the camera) to red (far from camera). Fine-tuning is performed using Stochastic Gradient Descent (SGD) with momentum of $\mu = 0.9$ and a batch-size of 128. The base learning rate is set to $\eta^0 = 0.01$ and reduced by $\eta^i = \eta^{i-1}0.99$ after each epoch. Similar to [18], data augmentation such as cropping and flipping are applied to extend the dataset. To that end, we resize images to 256×256 and draw cropping values from a discrete distribution in range $[0, 29]$. After fine-tuning RGB and depth encoders, we add and initialize the attention module and fusion layers, and similarly train f_{MAT} by SGD. We reduce the base learning rate to $\eta^0 = 0.001$ and train the network using a batch-size of 32. In case of both fine-tuning encoders and training the MAT, training runs for up to 100 epochs which takes 4-5 hours using an Nvidia GTX 1080 GPU.

At test time, we extract features X^9_{RGBD} from the last fully connected layer and use Euclidean distance to match features extracted in different camera views. Results are ranked according to the distance, intuitively, having the match with the shortest as the most similar. Since all datasets contain several images of each person, we apply a multi-shot approach and pool features extracted from all images of each person. Pooling is implemented by calculating average features which has previously shown superior to, e.g., maximizing when combining CNN features and a Euclidean distance metric [10, 15]. Although, in case of TVPR, we observe feature maximization to perform better and therefore provide results on that dataset using maximized features.

4.3 Experimental Results

We present results as Cumulative Matching Characteristic (CMC) curves that are produced by calculating a cumulative score for each *rank-i* indicating the number of persons having their true match within the *i* most similar in the ranked list.

The CMC curves produced from results on DPI-T, TVPR and OPR are shown in Figure B.3, along with results without the use of attention, similar to the proposed method of [18][1]. RGB_{att} and D_{att} represent the attention-based multilevel color and depth features X^8_{RGB} and X^8_D, while D-CNN, RGB-CNN and RGB-D-CNN represent the baseline depth, color and joint models, respectively.

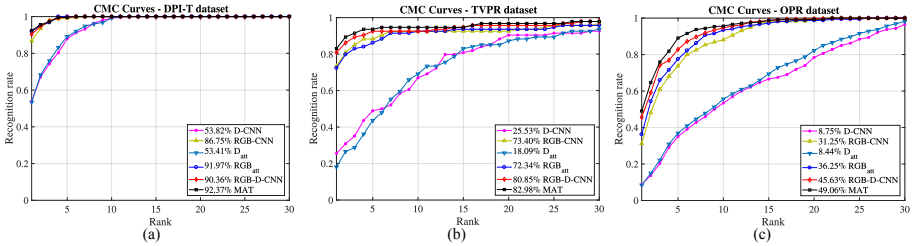


Fig. B.3: Experimental results on (a): DPI-T (p=249), (b): TVPR (p=94) and (c): OPR (p=32) for our multilevel attention-based RGB and depth features (D_{att} and RGB_{att}) along with MAT, and D-CNN, RGB-CNN and RGB-D-CNN proposed in [18].

From Figure B.3 it is clear that addition of attention-based features increases the rank-1 accuracy, even though, the accuracy is already high. Additionally, fusion of RGB and depth features outperform the use of RGB or depth individually. This is the case for DPI-T where the MAT increases the rank-1 accuracy by 2.01% and 0.4% compared to RGB-D-CNN and RGB_{att}, respectively. It is also worth noticing the increase of 5.22% when comparing RGB_{att} and RGB-CNN which shows the effect of using attention maps to extract local features and fuse those with global features. Similarly on TVPR, the rank-1 accuracy is increased by 2.13% and 10.64% compared to RGB-D-CNN and RGB_{att}, respectively. Comparing RGB_{att} and D_{att} to RGB-CNN and D-CNN, respectively, the use of attention does not seem to have a positive impact which could be due to misalignment issues from the detection, leading to missing information. This will be further analyzed in Subsection 4.5. Nonetheless, fusing the attention-based features results in a higher accuracy when comparing MAT and RGB-D-CNN. Finally on OPR, the MAT increases rank-1 accuracy by 3.43% and 12.81% compared to RGB-D-CNN and RGB_{att},

[1]In the original study, the authors identified a minor error in the input of OPR after publication, hence, results differ from those reported in [18].

respectively. Similar to DPI-T, fusing local and global information increases rank-1 accuracy by 5.00% when comparing RGB_{att} and RGB-CNN.

4.4 Analysis of Attention

To identify the contribution from the attention model, we visualize examples of attention maps α^4_{RGB}, α^5_{RGB}, α^4_D and α^5_D for all evaluated datasets. The visualizations are shown in Figure B.4. We show examples of persons having their true match as most similar ((a), (c) and (e)) and persons having their true match outside top-10 ((b), (d) and (f)). In case of TVPR and OPR, we randomly sample an image from each view and calculate attention maps. Since DPI-T consists of multiple sequences of each person, we randomly sample images from the most similar sequences between views.

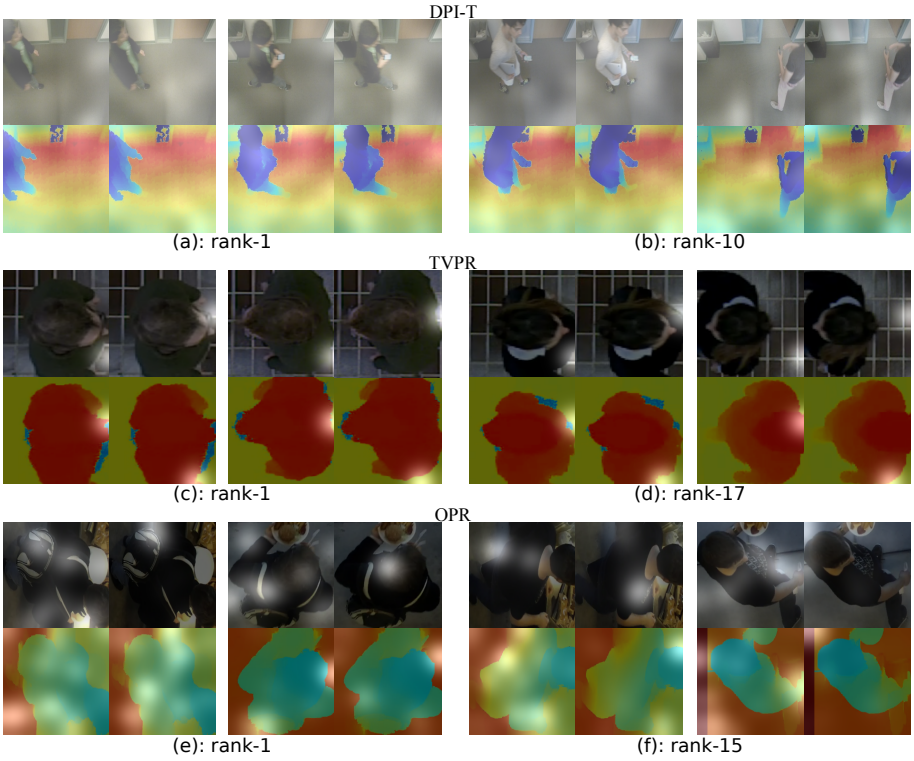


Fig. B.4: Examples of calculated attention maps in case of DPI-T, TVPR and OPR. Each subfigure consists of four attention maps from the same person in two different views. The four attention maps are organized as follows; top-left corner: α^4_{RGB}, bottom-left corner: α^4_D, top-right corner: α^5_{RGB}, bottom-right corner: α^5_D. Brighter areas means higher attention weights.

Generally, attention maps differs between the datasets. In case of DPI-T,

attention is mostly focusing on parts of the floor, although, attended regions also include parts of the person. This is most notable in case of α^4_{RGB} in Figure B.4 (a) where attention is mostly centered around the person and edges of the images. The pattern of α^5_{RGB} is more random, almost only capturing features from the floor. This behavior could be due to the encoding of depth images resulting in larger gradient changes in the floor compared to the persons, causing the floor to have a higher impact on the RGB based attention maps. Meanwhile, attention maps α^4_D and α^5_D focus on minor local regions centered around the floor. Considering full-frame images, combined with uniform colors of the scene, depth based attention maps are more affected by the colors of the floor, causing local features to almost not contain any information from the persons. This results in addition of noisy information, decreasing accuracy which is also clear when comparing D_{att} and D-CNN in Figure B.3 (a). This indicates the importance of extracting the ROI around the persons to remove as much background information as possible. In order to identify contributing regions, calculated attention maps before and after training the MAT should be compared. This will be considered in future work.

The attention maps for TVPR are less random but more similar across persons. In case of both Figure B.4 (c) and (d), α^4_{RGB} and α^5_D capture local information from the bottom right part of the images while α^5_{RGB} and α^4_D capture information in the center right part of the images. This cause images with misaligned detections to capture local features from the floor, in some cases, negatively affecting accuracy, as also shown in Figure B.3 (b) when comparing D_{att} and RGB_{att} to D-CNN and RGB-CNN. A reason for attention maps to be concentrated at the edges of the images could be the low resolution of depth information which results in useful gradient information only at the edge of the person. Although, in most cases, discriminative local information is extracted leading to a higher accuracy when fused with complementary global features.

Attention maps calculated in case of OPR are more centered around useful information. Comparing RGB based attention maps α^4_{RGB} and α^5_{RGB}, both mostly focus on the clothing which, typically, provide more discriminative information compared to, for example, the hair. Nevertheless, they still focus on different parts of the image, while α^4_{RGB} focus on multiple local regions with corresponding gradient changes in the depth image, α^5_{RGB}, focus on a single region. Additionally, the impact of fusing RGB and depth is shown by the attended regions, mostly centered near regions with larger gradient changes, for example, at the shoulder. This has a positive impact since these areas can be assumed to contain more useful information, considering the overhead view. α^4_D and α^5_D are more view dependent, focusing on several regions in the first view, while only focusing on a couple regions in the second. Like DPI-T, this could be due to a more diverse background in the first view. They both capture information around regions with larger gradients,

indicating that the attention model learns to calculate depth based attention maps that capture regions with useful color information while still preserving gradient information. A few failure cases exist as seen in the second (right) view of Figure B.4 (f). Here, a large gradient change in the left part of the depth image greatly affects the calculation of attention maps, causing attended regions to be centered around this edge. This is most likely a product of the depth calculations in [18] and should simply be removed in future evaluations.

4.5 Comparison to State-of-the-art

We compare our results with state-of-the-art for the three evaluated datasets. Due to the novelty of these datasets, only few results previously have been presented, including the 4D Recurrent Attention Mechanism (4D RAM) [17] and recurrent network with temporal attention (Depth ReID) [26] in case of DPI-T, and TVDH [19] in case of TVPR. Finally, the results of RGB-D-CNN$_{avg}$ (RGB-D-CNN) presented in [18] are compared. The comparisons are summarized in Table B.1-B.3, in all tables, "–" indicate non present results.

Method/Rank	r = 1	r = 5	r = 10	r = 20
4D RAM [17]	55.60	–	–	–
Depth ReID [26]	77.50	96.00	–	–
RGB-D-CNN [18]	90.36	**99.60**	**100**	**100**
MAT (ours)	**92.37**	**99.60**	**100**	**100**

Table B.1: Comparison between MAT and state-of-the-art systems on the DPI-T dataset (p=249). Best results are in bold.

Method/Rank	r = 1	r = 5	r = 10	r = 20
TVDH* [19]	75.50	87.50	89.20	91.90
RGB-D-CNN [18]	63.83	89.36	93.62	**97.87**
RGB-D-CNN†	80.85	92.55	92.55	95.74
MAT (ours)	**82.98**	**93.62**	**94.68**	96.81

Table B.2: Comparison between MAT and state-of-the-art systems on the TVPR dataset (p=94). Best results are in bold. (*Results are estimated from the CMC curve, †Reproduced by training and testing on images from detection).

Comparisons in Table B.1 show the MAT to outperform previously proposed methods. While 4D RAM and Depth ReID only consider depth information, RGB-D-CNN also considers color, showing the importance of fusing color and depth information. As also mentioned in Subsection 4.3 the MAT still increases accuracy, indicating the importance of including local discriminative features.

Method/Rank	r = 1	r = 5	r = 10	r = 20
RGB-D-CNN [18]	45.63	82.81	94.69	**99.69**
MAT (ours)	**49.06**	**89.06**	**95.62**	99.38

Table B.3: Comparison between MAT and state-of-the-art systems on the OPR dataset (p=32). Best results are in bold.

In case of TVPR shown in Table B.2, the MAT outperforms both RGB-D-CNN [18] and TVDH [19], increasing the rank-1 accuracy by 2.13% and 7.48%, respectively. Additionally, we note the importance of eliminating background noise which is shown by an increased rank-1 accuracy of 17.02% when comparing the original RGB-D-CNN results of [18] which considers full-frame images, and our evaluation using a similar system.

Finally, we compare the rank-1 through rank-20 accuracies, also depicted in Figure B.3, for the OPR dataset. Besides the rank-1 increase of 3.43%, the rank-5 accuracy is also greatly increased by 6.25% which is important to note, considering an image retrieval context where often the top-k most similar images are inspected by a person.

5 Conclusion

In this paper, we have proposed a Multimodal ATtention network (MAT) which implements an attention model with a multimodal CNN to calculate attention maps that capture local discriminative features from RGB and depth images. Attention maps are calculated by fusing RGB and depth information, resulting in attention maps that are calculated in a multimodal fashion. In total, four attention maps are calculated to extract local features from the fourth and fifth convolution layers of an RGB and depth CNN, respectively. Local RGB and depth based features are separately fused with global feature descriptors resulting in modality dependent multilevel features. Finally, multilevel RGB and depth features are fused to a multilevel RGB-D feature descriptor which better captures the correlation between RGB and depth information while including information at different abstraction levels. Evaluations on three overhead based datasets DPI-T, TVPR and OPR show the importance of fusing local and global information by increasing the rank-1 accuracy by 2.01%, 2.13% and 3.43%, respectively, compared to a similar network not considering attention.

To further increase accuracy, a more novel CNN should be considered while also the addition of an LSTM layer can be used to extend the network by additionally capture temporal information. By adding an LSTM, different attention modules can be considered, either spatial, temporal, or spatiotemporal.

Acknowledgement

This work is supported by Innovation Fund Denmark under Grant 5189-00222B.

References

[1] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[2] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person re-identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 1, pp. 1–20.

[3] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. ECCV*, 2016, pp. 17–35.

[4] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[5] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[6] Y. Zhang, B. Li, H. Lu, A. Irie, and X. Ruan, "Sample-specific svm learning for person re-identification," in *Proc. CVPR*, 2016, pp. 1278–1287.

[7] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[8] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. CVPR*, 2015, pp. 3908–3916.

[9] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proc. CVPR*, 2016, pp. 1335–1344.

[10] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-region bilinear convolutional neural networks for person re-identification," in *Proc. AVSS*, 2017, pp. 1–6.

[11] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[12] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, 2014, pp. 34–39.

[13] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *Proc. CVPR*, 2017, pp. 1077–1085.

[14] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[15] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proc. ECCV*, 2016, pp. 868–884.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. ICCV*, 2015, pp. 1116–1124.

[17] A. Haque, A. Alahi, and L. Fei-Fei, "Recurrent attention models for depth-based person identification," in *Proc. CVPR*, 2016, pp. 1229–1238.

[18] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[19] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, "Person re-identification dataset with rgb-d camera in a top-view configuration," in *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 1–11.

[20] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015, pp. 2048–2057.

[23] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[24] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[25] L. Zhao, X. Li, Y. Zhuang, and J. Wang, "Deeply-learned part-aligned representations for person re-identification," in *Proc. ICCV*, 2017, pp. 3219–3228.

[26] N. Karianakis, Z. Liu, Y. Chen, and S. Soatto, "Person depth reid: Robust person re-identification with commodity depth sensors," *arXiv preprint arXiv:1705.09882*, 2017.

[27] A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Enhancing person re-identification by late fusion of low-, mid- and high-level features," *IET Biometrics*, vol. 7, no. 2, pp. 125–135, 2018.

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

[29] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. CVPR*, 2017, pp. 384–393.

[30] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-identification," in *Proc. CVPR*, 2014, pp. 144–151.

[31] R. Zhao, W. Oyang, and X. Wang, "Person re-identification by saliency learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 2, pp. 356–370, 2017.

[32] F. Pala, R. Satta, G. Fumera, and F. Roli, "Multimodal person reidentification using rgb-d cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 788–799, 2016.

[33] A. Wu, W.-S. Zheng, and J.-H. Lai, "Robust depth-based person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2588–2603, 2017.

[34] R. Memisevic, "Learning to relate images: Mapping units, complex cells and simultaneous eigenspaces," *arXiv preprint arXiv:1110.0107*, 2011.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. CVPR*, 2017, pp. 7263–7271.

[38] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Proc. IROS*, 2015, pp. 681–687.

Paper C

Person Re-identification Using Spatial and Layer-Wise Attention

Aske R. Lejbølle, Kamal Nasrollahi, Benjamin Krogh, and Thomas B. Moeslund

The paper has been published in the
IEEE Transactions on Information Forensics and Security, Vol. 15(1), pp. 1216–1231, 2019.

The layout has been revised.

Abstract

Person re-identification requires extraction of discriminative features to ensure a correct match; this must be done independent of challenges, such as occlusion, view, or lighting changes. While occlusion can be eliminated by changing the camera setup from a horizontal to a vertical (overhead) viewpoint, other challenges arise as the total visible surface area of persons is decreased. As a result, methods that focus on the most discriminative regions of persons must be applied, while different domains should also be considered to extract different semantics. To further increase feature discriminability, complementary features extracted at different abstraction levels should be fused. To emphasize features at certain abstraction levels depending on the input, fusion should be done intelligently. This work considers multiple domains and feature discrimination, where a multimodal convolution neural network is applied to fuse RGB and depth information. To extract multi-local discriminative features, two different attention modules are proposed: (1) a spatial attention module, which is able to capture local information at different abstraction levels, and (2) a layer-wise attention module, which works as a dynamic weighting scheme to assign weights and fuse local abstraction-level features intelligently, depending on the input image. By fusing local and global features in a multimodal context, we show state-of-the-art accuracies on two publicly available datasets, DPI-T and TVPR, while increasing the state-of-the-art accuracy on a third dataset, OPR. Finally, through both visual and quantitative analysis, we show the ability of the proposed system to leverage multiple frames, by adapting feature weighting depending on the input.

1 Introduction

Since the beginning of the new millennium, person re-identification (re-id) has seen increased interest in the research community as the topic is perceived as both difficult and important [1–4]. Identification and verification involve matching an unknown signature to a database of either a single known or multiple known signatures. Re-id is the task of matching an anonymous signature to a database of anonymous signatures to find a correct match. Within computer vision, this is accomplished by matching signatures, i.e., features of a person extracted from images or a video in one camera view to features of persons extracted from images or a video in another. Features can, for example, contain hand-engineered low-level color and texture information [4–8], which can be extracted from small body patches, body parts, or the entire body, or they can contain high-level information by encoding low-level features using sparse coding [9–11]. Features are matched using a predefined metric, such as Euclidean distance, although, to increase the accuracy of the system, supervised metric learning [5, 12–15] is often considered to maximize the distances between non-matching feature pairs and minimize

the distances between matching ones. In recent years, however, Convolutional Neural Networks (CNNs) [16–21] have become increasingly popular due to their ability to learn discriminative high-level features by combining feature learning and classification in an end-to-end training scheme.

Due to increased data requirements when training CNNs, larger re-id datasets have been published in the last few years [22–24]. These datasets are more realistic in terms of the number of deployed cameras and environmental changes between views. A common characteristic of these datasets is the viewpoint, which is primarily horizontal and allows occlusions and changing views, as shown in Figure C.1 (a). To eliminate these challenges, the position of the camera can be changed to a vertical (overhead) viewpoint, as shown in Figure C.1 (b). In this work, only data captured from an overhead viewpoint is considered. Changing the viewpoint does, however, also increase the probability of removing important textural information from either the clothing or the face of a person. To counter the decrease in visual information, feature discriminability can be increased by adding additional information from other modalities. In connection with the overhead viewpoint, we add depth information when devising novel features. Adding depth enables us to model the height or body part ratios of persons, which can be used to learn a multimodal feature representation based on both RGB and depth modalities.

Fig. C.1: (a) Example of a horizontal viewpoint with changing views causing differences in appearance and partly occlusion [22]. (b) Example of an overhead viewpoint eliminating issues in (a) [25].

Additionally, the use of local feature representations has shown to outperform global ones [4]. In case of hand-engineered features, this is done by sampling small-image patches, typically of size 10×10 pixels, and extracting features from each patch. In terms of a CNN, this can be achieved by learning part-specific networks, either by splitting the body horizontally into a predefined number of body parts [18, 20] or by using part localization algorithms [21, 26]. Another option is to apply a soft attention mechanism [27] with the purpose of extracting features from only a single, or few, discrim-

inative regions in the input image. Different from horizontally splitting the body or applying localization algorithms, the attention mechanism captures information from local regions based on a learning scheme. This is based on calculating a two-dimensional weight matrix, which works as a mask on the input, where each element represents a weight in the interval [0,1] and is learned through back propagation. This will be referred to as *spatial attention*, which is applied in [28] to determine the importance of spatial locations at different layers of a neural network based on fusion of RGB and depth features. Since different layers of a CNN produce features at different abstraction levels [29], features produced by spatial attention modules represent local context information at different abstraction levels. To take advantage of complementary low-, mid-, and high-level information, features at different abstraction levels are often fused, simply by concatenation. Concatenation of features causes all elements in the resulting feature vector to be weighted equally, which is inexpedient if features at one abstraction level contain noisy information. Furthermore, local features from certain layers might be unnecessary and finding the optimal combination of relevant local features is impractical and time consuming if a very deep neural network is implemented. Instead, features should be fused using a dynamic weighting scheme that considers relevance to properly weight local features in order to maximize accuracy.

In this work, we introduce a multimodal dynamic weighting scheme as a layer-wise attention module to weight the output features of several spatial attention modules, based on the input. Since we focus on images captured from an overhead viewpoint which, depending on the height of the camera position, results in a more narrow view, sufficient data of each person might not be captured to properly exploit video-based methods, such as Recurrent Neural Networks (RNN). As a result, in this work, we consider only image-based models to learn a multimodal representation. Given a CNN consisting of L convolution layers, each convolution layer, in practice, can be followed by a spatial attention module and thus produce features that contain local context information. Instead of simply concatenating the features, the layer-wise attention module dynamically apply weights, which are learned by a learning scheme, to each feature vector and summarizes the outputs to a single discriminative feature vector. Thus, we end up with a multi-local context feature vector, which is a weighted summary of local context features at different abstraction levels. To take advantage of complimentary local and global information, multi-local context features from each modality are fused with high-level global features; these are referred to as multi-level features. Finally, multi-level modality features from RGB and depth, respectively, are fused to a multi-level multimodal feature representation. In summary, the main contributions include:

- A layer-wise attention module used to dynamically assign weights to local context features at different abstraction levels, depending on the input.
- An analysis of the output of the spatial and layer-wise attention modules used to reason how the data affects the weighting of features at different abstraction levels.
- A demonstration that a combination of spatial and layer-wise attention in a multimodal context provides state-of-the-art accuracy on several datasets collected from an overhead viewpoint.

The rest of the paper is structured as follows. Related work is outlined in Section C.2, which is followed by a description of the proposed system in Section C.3, including the baseline architecture as well as spatial and layer-wise attention modules. In Section C.4, experimental results are presented along with ablation studies and an analysis of the proposed attention modules. In addition, a comparison between the proposed system and state-of-the-art systems is presented. Finally, the work is concluded in Section C.5.

2 Related Work

2.1 CNN in Person Re-Identification

Since the development of early CNNs for the purpose of re-id, part-based learning has been considered with the aim to capture more discriminative local features. Yi et al. [20] proposed a CNN consisting of three separate streams, each processing an image that is split into a similar number of overlapping parts. Part-based features are then fused in a fully connected layer before classification. A similar approach was proposed by Cheng et al. [17], who split the body into four parts to learn part-specific features that are fused with full-body features. Similar to [20], features are fused late in the network. A more sophisticated part-based model was proposed by Ustinova et al. [18], who trained three part-specific sub-networks; instead of using a single sub-network per body part, they trained multiple sub-networks, and fused part-based features from corresponding sub-networks by a bilinear operation to retain geometric information in the input image.

More recently, Zhao et al. [21] proposed an architecture that consists of a Region Proposal Network (RPN) to locate 14-body joints in order to extract seven sub-regions of the body. Part-specific sub-networks were trained based on each of the sub-regions, followed by a Feature Fusion Network (FFN). This approach fuses part-specific information in a pyramid structure. Part-specific learning by joint localization was also proposed by Li et al. [30]; rather

than applying an RPN, they applied a Spatial Transformer Network (STN) to locate head-shoulder, upper-body, and lower-body regions.

Instead of training part-specific sub-networks, Suh et al. [31] trained a part map extractor to capture features from different body parts. Combined with an appearance map extractor to capture appearance features, a bilinear pooling operator is applied to fuse the two feature types, resulting in a part-aligned feature representation. Finally, Sarfraz et al. [26] proposed an architecture that takes as input a 17-channel image, including RGB and 14 keypoint channels containing keypoint locations. Furthermore, a separate view predictor was trained to model view information by calculating weights as probabilities of being 'front', 'back', or 'side'.

The proposed part based models [17], [18], [20] are able to capture and fuse local information from different body parts, and by adding a view predictor [26], it is even possible to add invariance to rotational changes. However, this assumes that images are captured from a horizontal viewpoint. If an overhead viewpoint is considered, certain body parts will be less visible, which will make it more difficult to achieve a proper result.

2.2 RGB-D CNN Models

Multimodal RGB-D CNNs have been proposed to a variety of applications [32–35]. For object recognition, Eitel et al. [32] proposed a two-stream CNN to fuse high-level RGB and depth features by adding a fully connected layer late in the network. An RGB-D CNN was proposed in [33] for pose estimation by, in a similar manner, processing RGB and depth images individually and using fused RGB-D features to train an SVM to determine the pose of objects. McCormac et al. [35] proposed a semantic mapping network, where depth is added as a fourth channel in the input to train an RGB-D semantic segmentation network.

Within re-id, the majority of published work focus on either RGB or depth, while fusion of the two modalities is rarely considered. Hand-crafted RGB-D features were devised by Liciotti et al. [25], who fused low-level color features from HSV histograms with anthropometric features extracted in the depth domain. Hand-crafted features were also devised by Wu et al. [36], who fused rotation invariant Eigen-depth features with low-level patch-based color and texture features. In the case of deep neural models, Karianakis et al. [37] proposed a combined CNN and LSTM to learn spatiotemporal depth features based on low-level knowledge transfer between an RGB and depth CNN. Additionally, they exploited frame-level weights by adding a Reinforced Temporal Attention (RTA) module, which infers the importance of each frame in a sequence using a hard attention mechanism, which has previously been introduced for image captioning [38]. Additionally, they considered fusion of spatiotemporal depth features and RGB features extracted from a CNN that

was trained on upper body images of persons. To our knowledge, the only other model within re-id to consider RGB-D features from a CNN is the work of [39], in which RGB and depth images are processed by modality- based sub-networks, while corresponding features are fused by concatenation in fully connected layers late in the network.

While [39] does not consider attention to capture local context features, [37] applies a coarse frame-level attention mechanism that does not capture and weight local information. Our proposed system does not only consider fusion of global RGB and depth features, but it also adds an attention mechanism to dynamically fuse local context features to consider complementary multi-level features.

2.3 Attention in Person Re-identification

An increasing number of CNNs that apply attention are being proposed in the field of re-id. Inspired by attention, Zhao et al. [40] proposed an architecture that uses part map detectors to estimate two-dimensional weight matrices that are multiplied by an input to output part feature maps. Here, the part map detectors are implemented as 1×1 convolutions followed by a sigmoid activation. A Comparative Attention Network (CAN) was proposed by Liu et al. [41], in which attention is applied to dynamically capture 'glimpses', i.e., minor regions in the input, by calculating spatial weight matrices used as masks on the input image. The dynamic element is added using a Long-Short Term Memory (LSTM) layer, which considers the masked input using the mask at the previous time step, along with the previous hidden state, and outputs a weight matrix, which attends a different area in the input. Masks were also generated by Song et al. [42], who proposed a mask-guided contrastive attention model consisting of three streams; one that learns from the regular input, and two others that learn from a foreground body and background image. The image is segmented by considering an additional binary mask in the input; this is combined with an attention loss to guide the attention map generation used for segmentation. Si et al. [43] proposed a Dual ATtention Matching network (DuATM), which learns a dual attention mechanism to match aligned feature pairs from an input triplet. Distances are then aggregated by average pooling and used with a triplet loss to update the network weights.

Li et al. [44] considered the misalignment issue in the input by proposing a Harmonious Attention CNN (HA-CNN), which combines regional attention [45], spatial attention [46], and channel attention [47] to capture both fine-grained pixel information at global level and coarse discriminative regions at local level. In case of depth modality, Haque et al. [48] proposed a recurrent attention model (RAM), which combines a localization network to capture glimpses with a CNN+LSTM to extract spatiotemporal features from

the regions. Attention has been applied also in the context of video-based re-id [49]. This will not be described in detail in the present work as the focus is only on image-based models.

Despite being able to learn fine-grained masks that are applied to the input, [41] adds additional complexity to the model by implementing attention using an LSTM, while [42] requires additional binary ground truth masks during training. Meanwhile, [44] fuses features from different local regions simply by concatenation before propagating the fused feature to a fully connected layer, hence, it does not consider the importance of each local region. Finally, previous work consider either RGB or depth as input during model training.

To our knowledge, the only previous work to consider multimodal attention is the Multimodal ATtention network (MAT) [28]. In this work, spatial attention weights are calculated for different layers of a CNN based on fusion of features from different modalities. Extracted local features are fused with global ones, and, finally, RGB and depth features are fused to a multimodal feature in the last layer of the network.

2.4 Dynamic Feature Fusion

Dynamic feature fusion has been studied mostly in connection with the fusion of multiple modalities [50–52]. To describe videos, Zhang et al. [50] proposed a combination of appearance and motion features from video clips that are dynamically fused by a weighted summary, where weights are calculated by applying an attention mechanism. The attention mechanism takes as input the motion or appearance feature, along with the intermediate hidden state from a decoder LSTM, to model the relevance of the feature. In video classification, Long et al. [52] proposed an Attention Fusion scheme in which RGB, flow, and audio features are fused by applying a Bidirectional LSTM, which models dependencies between modalities and, based on this, output global modality-based representations that are fused by concatenation. Attention-based dynamic feature fusion was also proposed for video description by Hori et al. [51], who applied a soft attention scheme using the previous hidden state of an LSTM decoder along with a modality feature to output a multimodal attention weight. Our proposed dynamic weighting scheme is mostly similar to the work of [51]. However, rather than dynamically fusing features from different modalities, we leverage multimodal information to fuse local abstraction level features for each modality.

Dynamic fusion of features at different abstraction levels has not often been considered [53, 54]. In case of human pose estimation, Chu et al. [53] proposed an 8-stack hourglass network, where each stack outputs multi-resolution attention maps that are fused by summation and applied to the output of the stack. Furthermore, Chen et al. [54] proposed an RGB-D object

detection network by introducing an Attention-ware Cross-modal Cross-level Fusion (ACCF) module, which concatenates RGB and depth feature maps and calculates channel-wise weights to model dependencies between RGB and depth channels. By propagating the output of an ACCF module late in the network back to lower layers of the network, predictions are generated in a coarse-to-fine manner.

In re-id, feature fusion most often is done by concatenation, as described in Section 2.2. In [44], local information generated from soft attention mechanisms is fused with global information generated from a hard attention mechanism; this is done by tensor addition to increase interaction. Lastly, Chang et al. [55] proposed the Multi-Level Factorization Net (MLFN), which consists of multiple blocks at different abstraction levels, each calculating a weighted summary of outputs from sub-networks in the given block. Weights from all blocks are additionally fused with features from the last block by an average operation. The former model only leverages RGB information to calculate attention weights, whereas the latter models use multi-level semantics by averaging multi-level features with a high-level feature representation. Here, we consider multimodal features to dynamically model and fuse features at different abstraction levels.

3 Proposed System

The proposed system is shown in Figure C.2. Given a pair of RGB and depth images, the system extracts multi-local context features by dynamically assigning weights to local context features at different abstraction levels. This is achieved by implementation of two attention modules: one that models the importance of spatial locations within feature maps at different abstraction levels (S-ATT), and another that models the importance of abstraction-level features (L-ATT). The output of the L-ATT is a feature vector containing local discriminative information, which is fused with a feature vector containing global information; this results in multi-level RGB and depth features. The two modality-based multi-level features are fused to generate a multimodal feature vector that is used for re-identification. The entire system is summarized in Table C.1; superscripts are neglected for simplicity. In the remaining part of the paper, we will refer to this system as *SLATT*.

3.1 Baseline Network Architecture

Similar to the work of [28], the backbone of the SLATT is an AlexNet CNN [56]. Following this architecture, the network consists of five convolution layers and three fully connected layers, where the first two fully connected layers transform features to sparse high-level representations, while the third

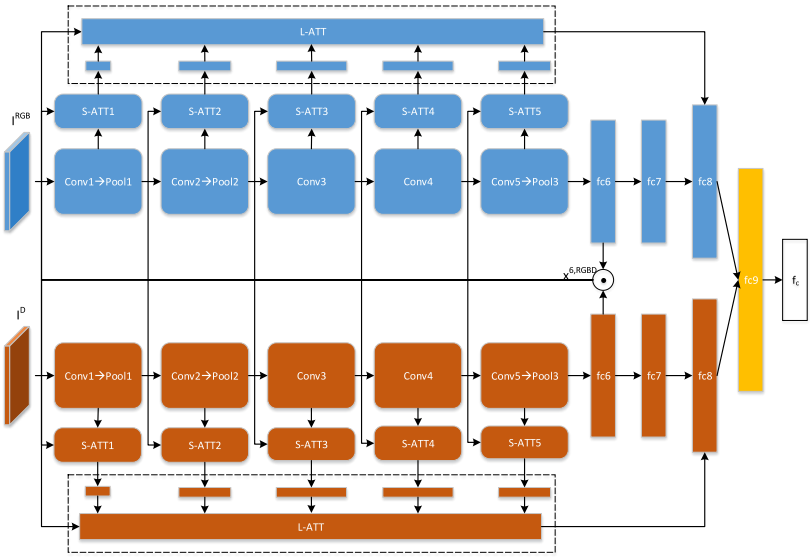


Fig. C.2: Overview of the Spatial and Layer-wise ATTention network (SLATT). RGB and depth images, respectively, are fed to separate CNNs pre-trained on modality dependent data. Each convolution layer in the network forward propagates the output, both to the next layer in the network and to S-ATT modules that calculate spatial attention features using RGB- and depth-based features, $x^{6,RGBD}$. Here, $x^{6,RGBD}$ is the resulting feature from fusing the rgb and depth features of the sixth network layer (fc6), respectively, by multiplication. This is indicated by the $\odot$ symbol. Outputs of S-ATT modules are propagated to the L-ATT module, which calculates attention-based feature weights. Modality-based local and global features are fused in $fc8$, while multi-level RGB and depth are fused in $fc9$. Finally, classification is performed in f_c.

fully connected layer acts as a classification layer. As part of the AlexNet, convolution layers one, two, and five are followed by max pooling layers to down-sample features and increase robustness to small translations, while Rectified Linear Units (ReLU) are used as activations. To increase the generalization, AlexNet introduced Local Response Normalization (LRN) before the activation and max pooling layers. However, since the introduction of Batch Normalization [57], which has shown to increase model accuracy and reduce training time, the LRN has become deprecated. Therefore, we remove the LRN layers and instead apply batch normalization. Similar to the ResNet architecture [58], we apply batch normalization after each convolution layer, but before ReLU activation. As we consider two modalities, the SLATT contains two identical parallel CNNs; each of these is processing either an RGB or a depth image. To learn modality specific features, weights between these networks are not shared.

The input to the system is an RGB/depth image pair $\{I_m^{RGB}, I_m^D\}, 1 \leq m \leq M$ sampled from the m'th person, where M denotes the total number of

Layer	*Output size*	*S-ATT output size*
Input	227×227×3 (x2)	
Conv1	55×55×96 (x2)	
Pool1	27×27×96 (x2)	1×96 (x2)
Conv2	27×27×256 (x2)	
Pool2	13×13×256 (x2)	1×256 (x2)
Conv3	13×13×384 (x2)	1×384 (x2)
Conv4	13×13×384 (x2)	1×384 (x2)
Conv5	13×13×256 (x2)	1×256 (x2)
Pool3	6×6×256 (x2)	
fc6	4096 (x2)	
fc7	4096 (x2)	
L-ATT	1024 (x2)	
fc8	4096 (x2)	
fc9	4096	
f_c	M	

Table C.1: Overview of the SLATT architecture, including output sizes from the S-ATT modules at each abstraction level. *M* denotes the number of persons in the training set. Similar structures are used to process RGB and depth images, respectively, and output multi-level features of similar sizes from fc8. This is indicated by (***x2***).

persons in the training set. The images are processed by corresponding CNN models, resulting in two global feature vectors $\{x_g^{7,RGB}, x_g^{7,D}\} \in R^{4096}$ from 'fc7' of the SLATT, where the subscript g indicates that the feature is *global*. Next, for each modality, we extract local context features as described in the following.

3.2 Spatial Attention (S-ATT)

The Spatial Attention (S-ATT) module applies a soft attention mechanism similar to that used for image captioning in [38]. Given an input of size $N \times C \times H \times W$, where N is the batch size, C the number of channels, and $\{H, W\}$ the height and width, respectively, the method works by calculating a local context vector $\hat{x} = \sum_i \alpha_i x_i$, which is the weighted sum of all feature vectors at spatial locations $1 \leq i \leq J | J = HW$. As described in [38], weights α_i can be calculated either hard using a stochastic function or soft using a deterministic function. While the performance between the two variations is largely comparable, the latter is more widely used as it can be easily integrated into the rest of a deep neural network. For a more direct comparison with [28], we consider only soft attention in this work. Soft spatial attention is applied in case of both RGB and depth, although, for simplicity, we neglect RGB and D superscripts in the following description.

In soft attention, weights are calculated from a parametrized score function, which outputs the score between an input feature and a reference vector using weights that are updated along with the rest of the CNN. In case of spatial attention, we define the score function as:

$$e_{l,i} = w_{l,i}^T ReLU(W_{x,l,i} x_{l,i} + W_{l,c} x_c), \tag{C.1}$$

where $e_{l,i}$ is a scalar representing the score between a vector $x_{l,i}$ from layer $1 \leq l \leq L$ at spatial location i and reference vector x_c. $W_{x,l,i}$ and $W_{l,c}$ are parametrized matrices, while $w_{l,i}$ is a parametrized vector. To take advantage of multiple modalities, RGB and depth features from 'fc6' of the SLATT are fused, and the resulting RGB-D feature, $x^{6,RGBD} \in R^{2048}$, is used as reference vector in the S-ATT module. To capture correspondences between modalities, features are fused by multiplication to capture higher-order dependencies, and fed to a fully connected layer, resulting in a feature vector consisting of values $x_i^{6,RGBD} = \sum_{j=1}^{2048} w_{ij} x_j^{6,RGB} x_j^{6,D}$, where w_{ij} is learned through back propagation. Thus, spatial attention scores are based on the multimodal behavior of the SLATT.

Weights at each spatial location are calculated by normalizing $e_{l,i}$ using a softmax function, defined as:

$$\alpha_{l,i} = \frac{\exp(e_{l,i})}{\sum_i \exp(e_{l,i})}, \tag{C.2}$$

where $\alpha_{l,i}$ is the attention weight at spatial location i for the l'th layer. Finally, the local context vector for layer l is calculated as:

$$\hat{x}_l = \sum_i \alpha_{l,i} x_{l,i}, \tag{C.3}$$

where the length of the context vector depends on the number of feature maps for a particular layer of the network. For our model, the output sizes are provided in Table C.1. In Section 4.3, we conduct an ablation study, which shows the accuracy by extracting and fusing local context features from S-ATT at different layers of the SLATT.

3.3 Layer-wise Attention (L-ATT)

The introduction of spatial attention implies that local context features are extracted at different abstraction levels. Still, in the work of [28], these are fused simply by concatenation. By doing so, low-level features containing information about, for example, texture are weighted equally to more high-level features describing larger parts, such as accessories. This is not expedient in case of an input with uniform textures and colors, or where different persons carry accessories that are similar in appearance. Instead, we propose that

each local context feature is weighted depending on the input. Thereby, we accomplish a more dynamic fusion scheme, which learns to consider feature importance in relation to the overall accuracy of the system. The dynamic weighting scheme is referred to as layer-wise attention (L-ATT).

As the number of feature maps differs between S-ATT modules, and the L-ATT requires features to be of same size, they are first aligned. This is accomplished by a transformation, $T : R^p \rightarrow R^q$, where p is the size of the feature, i.e., the number of feature maps, while q is the size of the aligned feature. To that end, a linear transformation is applied, which is defined by $\tilde{x}_l = W_l \hat{x}_l + b_l$. In our network, this is implemented using a fully connected layer, where W_l and b_l are the weight and bias, respectively, that are learned along with the rest of the network during training. In Section 4.3, an ablation study is conducted by varying the size of q.

The proposed layer-wise attention module follows an approach similar to that for the S-ATT modules. Given K local context feature vectors, weights β_l are calculated from the scores between the features and a reference. Similar to (C.1), we can define a score function as:

$$a_l = w_l^T ReLU(W_{\tilde{x},l} \tilde{x}_l + W_c x_c), \tag{C.4}$$

where a_l is the score represented as a scalar, $\tilde{x}_l$ is the aligned local context feature, while w_l, $W_{\tilde{x},l}$, and W_c are parametrized vectors and matrices that influence how the particular feature is weighted. Likewise, weights β_l are calculated by softmax normalization:

$$\beta_l = \frac{\exp(a_l)}{\sum_l \exp(a_l)} \tag{C.5}$$

Finally, the weighted sum of local context features is calculated as:

$$x_{lo} = \sum_l \beta_l \tilde{x}_l, \tag{C.6}$$

where the subscript lo indicates the feature containing *local* information.

In this work, the layer-wise attention is applied using an AlexNet architecture as backbone, as described in Section 3.1. In principle, the module can be applied to any combinations of features at different abstraction levels and using any network, such as ResNet [58] or GoogLeNet [59].

Multi-level features are learned for each modality by adding a fully connected layer 'fc8', which takes as input the concatenated global and local features, $x_{gl} = [x_{lo}, x_g]$, and outputs a multi-level feature, x_{ml}. The multi-level modality features x_{ml}^{RBG} and x_{ml}^{D} are fused and used as input to an additional fully connected layer, 'fc9', which outputs a multi-level multimodal feature, x_{ml}^{RGBD}, used for classification.

Classification is implemented as a fully connected layer followed by a softmax layer, calculating the probability of a person belonging to the correct class. Given the feature $x_{ml,m}^{RGBD}$, calculated from the input image pair $\{I_m^{RGB}, I_m^D\}$, the probability is calculated as: Along with the true label, m, the logistic loss function is used to calculate the error over the entire batch of size N, defined as:

Classification is implemented as a fully connected layer followed by a softmax layer. Given an input pair, $\{I_m^{RGB}, I_m^D\}$, the probability of a person belonging to the correct class, given the feature $x_{ml,m}^{RGBD}$, is defined as $\hat{p}_i = Pr(y = m|x_{ml,m}^{RGBD})$. Along with the true label, m, the logistic loss function is used to calculate the error over the entire batch of size N, defined as:

$$L_{id} = -\frac{1}{N}\sum_{i=1}^{N}\log(\hat{p}_i) \tag{C.7}$$

4 Experiments

Extensive experiments are conducted on three RGB-D datasets that are all collected from an overhead viewpoint. First, details on training of the SLATT are described in Section 4.1, which is followed by a description of the three evaluated datasets in Section 4.2. Ablation studies are presented in Section 4.3, and the results are used as basis in the experimental results in Section 4.4. A visual analysis is presented in Section 4.5, and the results are finally compared to state-of-the-art systems in Section 4.6.

4.1 Implementation Details

Training of the SLATT follows a two-step approach. First, modality-based CNNs are trained individually to adapt network parameters to the context of classifying persons within respective domains. In both cases, weights are initialized from a model pre-trained on the ImageNet dataset [56]. Training is performed using Stochastic Gradient Descent (SGD) with a base learning rate of $\eta^0 = 0.001$ and reduced by $\eta^i = \eta^{i-1}0.99$ after each epoch. To further accelerate the training, we add a momentum of $\mu = 0.9$ and train with a batch size of 128. To increase the amount of data and make the network more invariant to translational changes, common augmentation techniques, such as random cropping and flipping, are applied, and the data is shuffled before each new epoch. In case of cropping, images are resized to 256×256 pixels, and cropping values are drawn from a discrete distribution in the interval [0, 29]. To avoid overfitting and increase generalization, dropout is placed after layers 'fc7', 'fc8', and 'fc9' using probability values 0.5, 0.5, and 0.8, respectively. In case of training the depth-based CNN, depth images

need to be converted to an appropriate format to take advantage of the pre-trained ImageNet weights. To that end, a JET colormap is applied, which encodes each depth value to an RGB value; red represents objects that are far away, whereas green to blue, represent objects that are close. Applying a JET colormap is fast and has previously shown to outperform other encoding techniques [32]. Weights from the trained RGB- and depth-based CNNs are used to initialize the convolution layers and the first two fully connected layers of the SLATT model. Weights of the remaining layers are initialized using values drawn from a Gaussian distribution with zero mean and a standard deviation of $\sqrt{1/in\_size}$, where *in_size* refers to the number of input neurons. Hyper-parameters, which are similar to those used to train RGB and depth CNNs, are used to train the SLATT; in both cases, the training runs for 100 epochs. Training is performed on an Nvidia GTX 1080 and takes up to 1.5 hours for modality-based CNNs and up to 4 hours in case of SLATT.

At test time, multimodal features from 'fc9' of the SLATT are extracted from images of persons captured in different camera views. We follow a multi-shot approach and extract features from all images of each person. Features are then summarized by average pooling. Euclidean distance is calculated between all pairs of persons across views and sorted by distance. Thus, shorter distances indicate increased similarity between pairs.

4.2 Datasets and Protocols

Evaluation of the SLATT is performed on three datasets: *Depth-based Person Identification from Top* (DPI-T) [48], *Top View Person Re-identification* (TVPR) [25], and *Overhead Person Re-identification* (OPR) [39]; the two former are publicly available. To our knowledge, these are the only RGB-D based re-identification datasets collected from an overhead viewpoint.

DPI-T This dataset consists of 12 persons captured in an average of five appearances in a hallway. An average of 25 sequences are recorded of each person. These are split into 213 training sequences and 249 test sequences. At test time, all test sequences are matched against all training sequences.

TVPR This dataset contains recordings of 100 persons appearing twice in a hallway; first walking from left to right and then from right to left. Sequences of the first appearance make up the training set, while those of the second appearance constitute the test set. Similar to DPI-T, during tests, all test sequences are matched against all training sequences. For better comparison with [28], we consider the same 94 of the 100 persons, while also doing the evaluation on Region of Interest (ROI) images that are extracted using the

You Only Look Once (YOLO) detector [60][1].

OPR This dataset contains sequences of 64 persons captured in a canteen area. Each person appears twice; when entering the canteen and again when leaving the canteen. In contrast to DPI-T and TVPR, the evaluation of this dataset follows a protocol that is commonly known from RGB-based datasets, such as Market-1501 [9] or CUHK03 [61]. This implies that the data is randomly split into training and test sets, each containing 32 persons. At test time, re-id is performed on the 32 unseen persons. Additionally, 10 random training/test splits are performed, and the average accuracy is calculated across all 10 iterations.

4.3 Ablation Studies

An ablation study is conducted by configuring the number of considered local context features when only spatial attention is applied. From an empirical study, in [28], only the outputs from S-ATT4 and S-ATT5 are considered. In this work, more extensive experiments are conducted in order to show the impact on accuracy when either adding or removing local context features from additional S-ATT modules.

Table C.2 shows the impact of adding additional local context features at different abstraction levels, starting by only considering the output from only global features and incrementally adding features from S-ATT5 down to S-ATT1. In this case, similar to [28], features are fused by concatenation. Tests are conducted on the datasets presented in Section 4.2 and follow the training protocols described in Section 4.1.

S-ATT1						✓
S-ATT2					✓	✓
S-ATT3				✓	✓	✓
S-ATT4			✓	✓	✓	✓
S-ATT5		✓	✓	✓	✓	✓
OPR Rank-1:	59.38	64.69	63.44	64.38	**65.63**	64.06
DPI-T Rank-1:	94.38	97.19	97.19	**97.59**	97,19	96.79
TVPR Rank-1:	92.55	**94.68**	**94.68**	**94.68**	93.62	93.62

Table C.2: Impact on rank-1 accuracy by changing the number of S-ATT modules in fusion of local context features. Best result in each dataset is highlighted in bold.

Contrary to [28], the best results do not only include the outputs from S-ATT4-5, but rather the outputs from S-ATT2-5 or S-ATT3-5. Since [28] does

[1] Annotations and ROI extraction guide provided at: `https://github.com/Lejboelle/TVPR_annotations`

not consider batch normalization, the results are not entirely comparable, but they still provide a good indication of the relevance of feature types across different datasets. In case of OPR, features from S-ATT2 complement additional local and global features, while this is not the case for DPI-T and TVPR, where accuracy is decreasing if additional features from S-ATT1-2 are included. This could be due to the original resolution of the images in OPR, which is higher and thus enables capture of more detailed information at a lower abstraction level. However, overall we see an increase from adding local context features, which shows the benefit from the S-ATT modules.

Next, an ablation study is conducted by varying the feature size of the L-ATT module. This impacts both the size of the aligned features, $\tilde{x}_l$, and the size of the output feature, x_{lo}. Table C.3 summarizes the results. In case of DPI-T and TVPR differences are marginal between feature sizes of 256 and 1024, while in case of OPR, a feature size of 1024 increase accuracy by 2.18% and 3.44%, respectively, compared to 512 and 256.

	Feature size		
	256	*512*	*1024*
OPR Rank-1:	63.12	64.38	**66.56**
DPI-T Rank-1:	**96.79**	95.98	96.39
TVPR Rank-1:	**94.68**	93.62	93.62

Table C.3: Impact on rank-1 accuracy by changing the size of x_{lo}. Best result in each dataset is highlighted in bold.

4.4 Experimental Results

Based on Table C.3, the following results of the SLATT are based on a feature size of $x_{lo} \in R^{1024}$. Results are presented as Cumulative Matching Characteristic (CMC) curves, that is, for each rank-i, a cumulative score is calculated, which represents the percentage of persons having their truth match within the i most considered. The results are compared with application of only spatial attention, in this case consideration of S-ATT3-5 (MAT35), as appears from the results in Table C.2, but also of S-ATT1-5 (MAT15), which provides a more direct comparison when additional layer-wise attention is applied. Furthermore, results are compared with the baseline RGB-D-CNN architecture [39] without attention to show the benefit of fusing global and local information. CMC curves showing accuracies on OPR, TVPR, and DPI-T are shown in Figure C.3 (a), (b), and (c), respectively. In case of TVPR, the RGB-D-CNN network is able to re-id almost all persons in the dataset. Since this data was acquired in controlled environmental settings, the only real challenge is the rotational change from walking horizontally in both directions. Improving this result is, therefore, a difficult task. Nonetheless, MAT35 in-

creases the accuracy by 2.13% compared to RGB-D-CNN. Thus, adding local features increases the overall accuracy, although low-level features from S-ATT1 and S-ATT2 do not add additional discriminative information, as also seen in Table C.2. Nevertheless, it is also worth noting the rank-2 accuracy, which is similar between RGB-D-CNN and MAT35. This indicates the importance of including local features to distinguish between persons with much similar appearance.

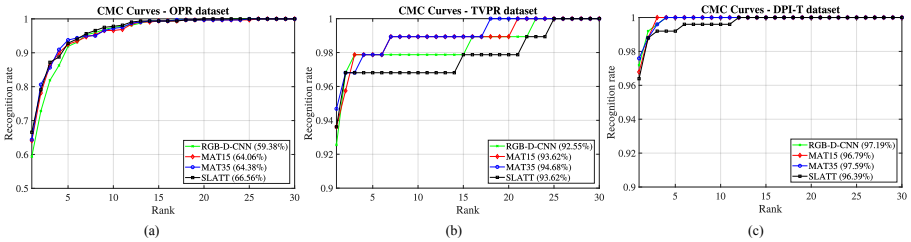


Fig. C.3: CMC curves based on experimental results on (a) OPR (p=32), (b) TVPR (p=94), and (c) DPI-T (p=249).

A similar result is seen in case of DPI-T, where the results of MAT35 and SLATT are almost identical, where the SLATT, in this case, is inferior to MAT35. Since this dataset consists of only 12 persons, where several sequences are captured of each person, the accuracy of this dataset also seems to be saturated at 97.59% and is therefore difficult to increase. Due to saturated accuracies both on DPI-T and TVPR, we analyze the contribution of the L-ATT module by comparing single-shot and multi-shot accuracies in Subsection 4.7.

The results on the more challenging OPR dataset clearly show the benefit of weighting local features dynamically. While MAT15 shows the smallest increase in accuracy of 4.68% compared to RGB-D-CNN, MAT35 increases the accuracy by 5.00%, while SLATT shows an increase of 7.18%.

4.5 Visual Attention Analysis

To obtain a better understanding of the relevance of local context features at different abstraction levels, we visualize spatial attention maps from S-ATT modules, which will henceforth be referred to as S-ATT maps, along with their corresponding L-ATT weights. The goal of this analysis is twofold: (1) to identify which local context features are captured at each abstraction level, and (2) to identify trends in the dynamic weighting of features in relation to the dataset. We show examples of success cases to identify discriminative feature regions that result in correct re-identification. Examples are shown for all datasets presented in Section 4.2 by randomly sampling four images

from a person in each dataset and calculating S-ATT maps along with L-ATT weights. Figures C.4, C.5, and C.6 show examples of calculated weights in case of OPR, TVPR, and DPI-T, respectively. Each row shows the S-ATT maps from a single layer, going from S-ATT1 at the top to S-ATT5 at the bottom. RGB-based S-ATT maps are shown to the left, while the depth-based ones are shown to the right. Above the S-ATT maps, layer-wise weights are shown.

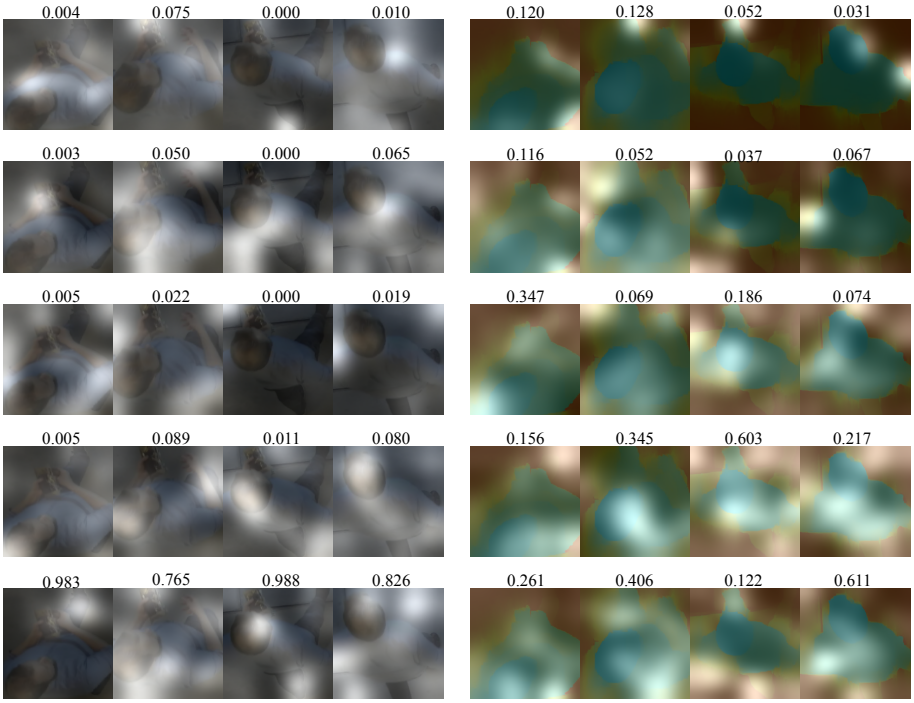


Fig. C.4: Visualization of RGB-based (left) and depth-based (right) S-ATT maps with corresponding L-ATT weights for the OPR dataset. Each row shows S-ATT maps from four randomly sampled images of the same person; the first row shows the output from S-ATT1 down to S-ATT5 at the bottom.

The RGB-based S-ATT maps that are shown in case of OPR in Figure C.4 indicate a trend to mostly weight the output of S-ATT5, which is the case for all four images. Even though S-ATT5 is highly weighted, differences in S-ATT maps are seen. While the first image captures information around the legs, the second one captures information at the head and shoulder regions, while the third highlights head and legs. More diverse L-ATT weights are seen in case of depth images. S-ATT maps generally tend to highlight regions around the edges, for example at the head/shoulders or around the entire body. While low-level S-ATT maps are mostly concentrated around a few points of interest, S-ATT maps at higher abstraction levels include larger edge

regions. A general trend is seen for S-ATT maps, but the L-ATT module is able to dynamically weight features depending on the input, as shown by the differences across the four images. Although weights are distributed more evenly across layers, the outputs of S-ATT3-5 are generally weighted higher.

When the L-ATT weights in Figure C.5 are inspected, a trend similar to that in Figure C.4 is seen in case of RGB, where features at higher abstraction levels are weighted higher by the L-ATT module. The S-ATT maps show more similarities across the four images, where mostly the head and shoulders are highlighted. Nonetheless, the dynamic weighting causes different information to be fused by weighting low-level features higher in the first image compared to the three other images. Similar to Figure C.4, in case of depth, L-ATT weights are more evenly distributed, although features at lower abstraction levels are weighted higher. S-ATT maps are also more centered around few edge points across all layers, while, in case of OPR, this applies typically at lower abstraction levels.

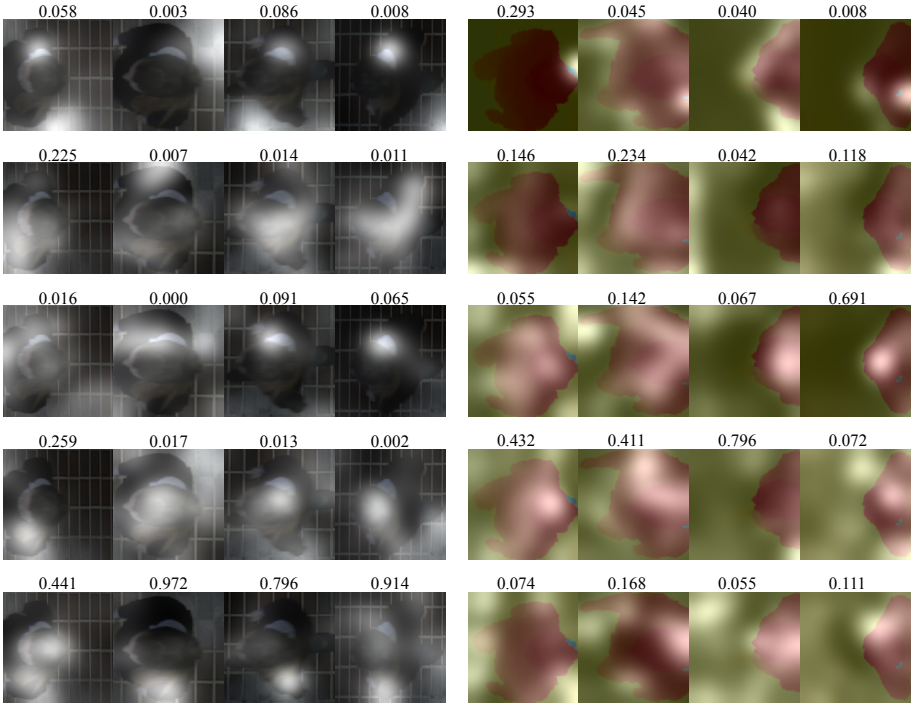


Fig. C.5: Visualization of RGB-based (left) and depth-based (right) S-ATT maps with corresponding L-ATT weights for the TVPR dataset. Each row shows S-ATT maps from four randomly sampled images of the same person; the first row shows the output from S-ATT1 down to S-ATT5 at the bottom.

Larger differences in RGB-based L-ATT weights are seen in Figure C.6. The first two images show more evenly distributed weights, whereas the last two mostly weight features at S-ATT5, but they still add complementary low-level information. Larger differences are also seen in case of the S-ATT maps, where both the legs, the frontal body, and the head are highlighted. Similar to OPR and TVPR, depth-based S-ATT maps are centered mostly around edges of the body. However, in contrast to the two former, the head is less highlighted. Likewise, L-ATT does not weight S-ATT1 or S-ATT5 higher, but it distributes weights at all abstraction levels more evenly.

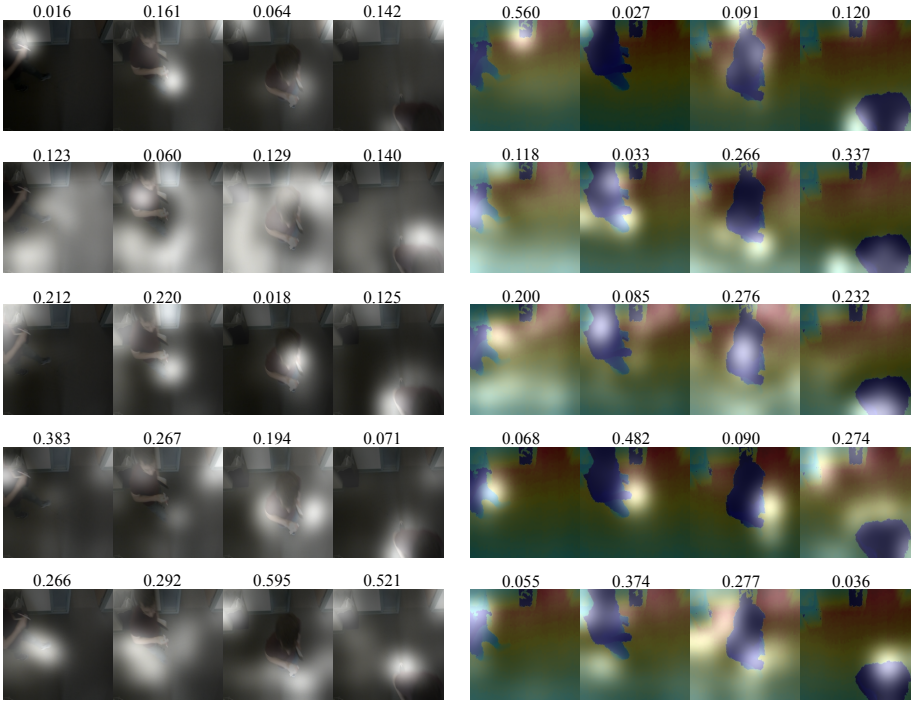


Fig. C.6: Visualization of RGB-based (left) and depth-based (right) S-ATT maps with corresponding L-ATT weights for the DPI-T dataset. Each row shows S-ATT maps from four randomly sampled images of the same person; the first row shows the output from S-ATT1 down to S-ATT5 at the bottom.

OPR and TVPR that are captured from a more vertical viewpoint and in a less complex scene compared to DPI-T generally weight higher local RGB information at higher abstraction levels, which could be due to less visible texture. This is also indicated by the RGB-based S-ATT maps, which highlight the head and shoulder regions. The depth-based S-ATT maps are more similar across all three datasets as they mostly highlight body edges. Still, while OPR and TVPR place higher weight on the mid- and higher-level

features, DPI-T, also in this case, weights low-level features. In all cases, the dynamic weighting scheme ensures fusing of the most relevant features, which are extracted at different abstraction levels. The differences in L-ATT weights across the dataset, which are especially clear when comparing OPR and TVPR to DPI-T, show the strength of the L-ATT to properly weight features at different abstraction levels depending on the data.

Finally, Figure C.7 shows cases of incorrect re-id to identify challenging issues in the SLATT. The L-ATT weights show similar trends as for correct re-id. Therefore, the issue lies in the input and the S-ATT maps. In case of

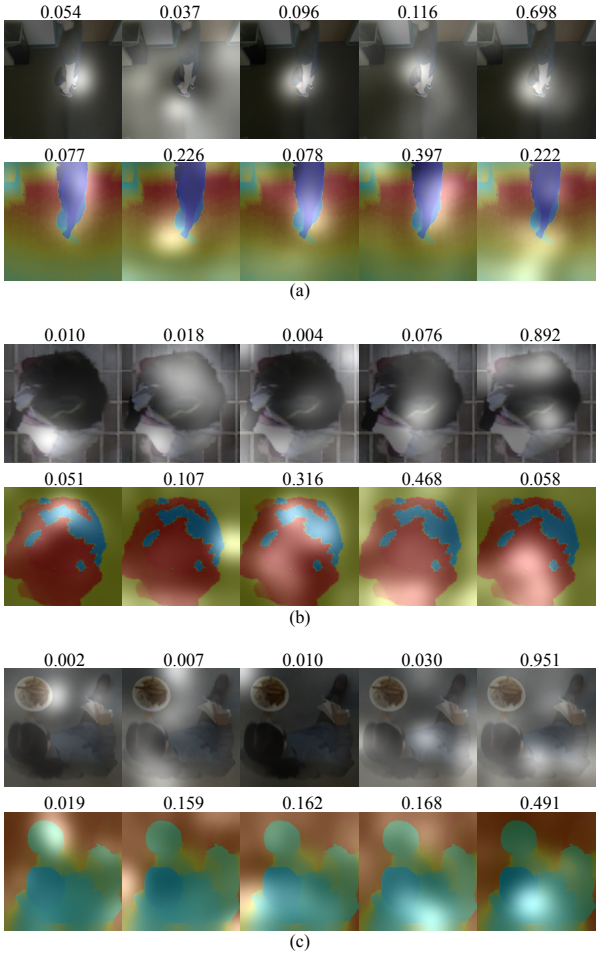


Fig. C.7: Visualization of S-ATT maps with corresponding L-ATT weights in failure cases for (a) DPI-T, (b) TVPR, and (c) OPR. Both RGB-based S-ATT maps (top) and depth-based attention maps (bottom) are shown from the S-ATT1 (left) to S-ATT5 (right).

both (a) DPI-T and (b) TVPR, RGB-based S-ATT maps are centered around few similar areas: the arm in case of the former, and the hairline in case of the latter. For DPI-T, the depth-based S-ATT maps also mostly highlight the arm, which causes redundant information to be fused. The depth images in (b) also show areas of undefined depth, which is indicated by blue regions, and this results in noisy information. In case of (c) OPR, the S-ATT5 map, which is by far weighted the highest, is quite sparse, and this causes capture of noisy information. Sparsity is also seen in S-ATT4 and S-ATT5 maps in Figure C.4. These are, however, different from this failure case as the less noisy S-ATT4 features in Figure C.4 are weighted higher. The depth-based S-ATT maps highlight more non-relevant areas, such as the plate or the floor. This is especially seen when inspecting S-ATT3 and S-ATT4 maps. This could indicate difficulties when a person carries objects that are common to the scene, in this case a plate of food.

4.6 Comparison with State-of-the-Art Systems

Comparisons between the results of the SLATT, presented in Figure C.3, and state-of-the-art systems are provided in Tables C.4-C.6.

Previously proposed systems have evaluated the DPI-T dataset using only depth information. As a result, we compare the results of our SLATT and previous RGB-D CNNs by extracting depth features, which is indicated by the subscript D. We compare the results with the residual attention (4D RAM) proposed in [48] and with CNN-LSTM (Depth ReID) proposed in [37] along with the RGB-D-CNN [39] and MAT [28], both with and without the use of batch normalization. MAT35$_D$ (ours) refers to the results of the MAT, which considers additional local context features from S-ATT3. Furthermore, we also provide comparisons of RGB-D-CNN, MAT, MAT35, and SLATT with RGB information included. In all cases, MAT35 and SLATT make use of batch normalization.

As seen in Table C.4, the use of batch normalization clearly increases the accuracy, which is shown for both baseline RGB-D-CNN and MAT. Moreover, including additional local information at lower abstraction levels decreases the accuracy when comparing MAT$_D$+BN and MAT35$_D$. This could indicate that low-level depth features do not provide enough discriminative information to ensure benefits. This could also be the reason why the SLATT$_D$ provide a rank-1 accuracy which is inferior to MAT$_D$+BN. However, the addition of RGB information increases the accuracy by up to 14.46% when comparing MAT$_D$+BN and MAT35 while SLATT provide accuracies almost similar to MAT35. Even though the accuracy is high when including only depth information, the complementarity of RGB and depth, combined with the use of both local and global features, produces more discriminative features, which results in higher accuracy.

Method/Rank	r = 1	r = 5	r = 10	r = 20
4D RAM [48]	55.60	–	–	–
Depth ReID [37]	76.30	–	–	–
RGB-D-CNN$_D$ [39]	53.82	87.95	99.20	100
RGB-D-CNN$_D$ [39]+BN	80.72	97.59	100	100
MAT$_D$ [28]	53.41	89.16	99.20	100
MAT$_D$ [28]+BN	**83.13**	97.19	100	100
MAT35$_D$ (ours)	81.93	97.99	100	100
SLATT$_D$ (ours)	79.52	97.59	100	100
RGB-D-CNN [39]+BN	94.38	99.20	100	100
MAT [28]+BN	97.19	100	100	100
MAT35 (ours)	**97.59**	100	100	100
SLATT (ours)	96.39	99.20	99.60	100

Table C.4: Comparison between SLATT and state-of-the-art systems on the DPI-T dataset (p=249) ('–' indicate that a result is not available). Best results are highlighted in bold.

Besides RGB-D-CNN and MAT, with and without batch normalization, the only other comparable system in case of TVPR, as seen in Table C.5, is the one of [25], where hand-crafted RGB-D features are extracted (TVDH). Similar to DPI-T, the addition of batch normalization results provides a significant increase in accuracy, while CNN-based features outperform the hand-crafted ones by up to 19.38% when comparing TVDH and MAT35. In contrast to DPI-T, additional information from S-ATT3 does not increase the accuracies when comparing MAT+BN and MAT35. When adding layer-wise attention, we do not benefit from additional low-level information and achieve a rank-1 accuracy similar to that of MAT35. This could be due to accuracy being close to saturated or the resolution of depth, which result in uniformly colored images after applying the JET color map.

Method/Rank	r = 1	r = 5	r = 10	r = 20
TVDH* [25]	75.50	87.50	89.20	91.90
RGB-D-CNN [39]	80.85	92.55	92.55	95.74
RGB-D-CNN [39]+BN	92.55	97.87	97.87	100
MAT [28]	82.98	93.62	94.68	96.81
MAT [28]+BN	94.68	97.87	97.87	97.87
MAT35 (ours)	**94.68**	97.87	97.87	100
SLATT (ours)	93.62	96.81	97.87	100

Table C.5: Comparison between SLATT and state-of-the-art systems on the TVPR dataset (p=94). Best results are highlighted in bold. (*Results are estimated from the CMC curve.

Comparisons between SLATT and state-of-the-art systems on the OPR dataset, which is provided in Table C.6, indicate that more importance should be directed towards dynamic feature weighting schemes when difficult datasets are being evaluated. As also seen in Table C.2, adding local features from S-ATT3 increases the accuracy by 0.94% when comparing MAT+BN and MAT35. The rank-1 accuracy is decreased to 64.06% when adding additional local features from S-ATT1 and S-ATT2, as shown in Table C.2, but dynamically weighting the features using layer-wise attention increases the accuracy by 2.50%. Additionally, compared to the previous work of [28] with BN, the rank-1 accuracy of the SLATT is increased by 3.12% while the accuracy is increased by 7.18% compared to RGB-D-CNN+BN.

Method/Rank	r = 1	r = 5	r = 10	r = 20
RGB-D-CNN [39]	45.63	82.81	94.69	99.69
RGB-D-CNN [39]+BN	59.38	91.88	97.50	99.69
MAT [28]	49.06	89.06	95.62	99.38
MAT [28]+BN	63.44	92.50	96.25	99.69
MAT35 (ours)	64.38	93.75	97.19	99.69
SLATT (ours)	**66.56**	92.81	**97.81**	**100**

Table C.6: Comparison between SLATT and state-of-the-art systems on the OPR dataset (p=32). Best results are highlighted in bold.

To further highlight the significance of the proposed system, we provide pairwise statistics of the rank-1 accuracy on OPR between SLATT and the three systems of MAT, MAT35 and RGB-D-CNN. A comparison is provided as a box plot in Figure C.8. From here, it is clear that the variety of MAT is lower than that of SLATT, however, the maximum observed value of SLATT is higher while the minimum observed value is higher than all three compared systems. Additionally, while the medians of MAT and MAT35 are higher than that of RGB-D-CNN, the median of SLATT is higher than all three.

In addition to Figure C.8, we also provide paired t-tests to show the significance in terms of probabilities. That is, between SLATT and the remaining three systems we calculate t-values using the differences in rank-1 accuracy between methods during all 10 test runs. Given the t-value, we use a look-up table to infer a corresponding p-value, which is an indicator of the level of significance. The t-value is calculate as:

$$t = \frac{\bar{d}}{SE(\bar{d})}, \tag{C.8}$$

where $\bar{d}$ is the mean of differences while $SE(\bar{d})$ is the standard error of the mean differences, calculated as $SE(\bar{d}) = \frac{s_d}{\sqrt{n}}$, where s_d is the standard

deviation of differences and n is the number of test iterations, i.e. 10 in our case.

Table C.7 provides an overview of pairwise t-values and corresponding p-values.

Observing the p-values of SLATT/MAT and SLATT/RGB-D-CNN in Table C.7, there is strong evidence that the inclusion of the L-ATT module results in higher accuracies since, in both cases, the value is less than 0.05. Compared to MAT35, the results are marginally significant since the p-value is just above 0.05, which still indicates good evidence of a positive impact on accuracy.

	SLATT/MAT35	SLATT/MAT	SLATT/RGB-D-CNN
t-value	2.091	3.001	3.977
p-value	0.066	0.015	0.003

Table C.7: p- and t-values from pairwise t-tests between SLATT and MAT, MAT35 and RGB-D-CNN, respectively.

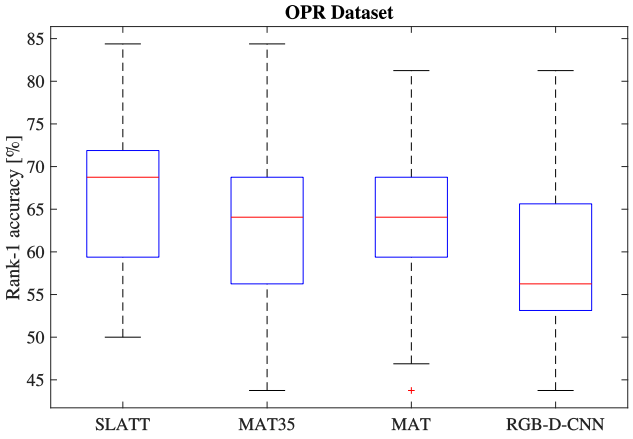


Fig. C.8: Comparison of statistical differences between SLATT, MAT35, MAT and RGB-D-CNN.

4.7 Contribution of L-ATT

From the visual analysis in Section 4.5, it is clear that the L-ATT module is able to dynamically weight features at different abstraction levels based on the input. To further study the effect of this property, we compare the results of the multi-shot setting with a single-shot setting, where only a single image of each person is available at the time of testing. In the single-shot setting,

we randomly sample an image from each person in both camera views and, similarly to the multi-shot setting, calculate Euclidean distances between extracted features. In both settings, we consider only the rank-1 accuracy and compare the relative increase from single- to multi-shot accuracy across RGB-D-CNN+BN, MAT+BN, MAT35 and SLATT. Table C.8 provides an overview of rank-1 accuracies in case of single- and multi-shot settings, respectively, while Figure C.9 shows the relative increase between the two settings.

	OPR		TVPR		DPI-T	
Method/Rank-1	Single-shot	Multi-shot	Single-shot	Multi-shot	Single-shot	Multi-shot
RGB-D-CNN [39]+BN	46.25	59.38	76.60	92.55	93.17	94.38
MAT [28]+BN	43.44	63.44	80.85	94.68	90.76	97.19
MAT35 (ours)	43.44	64.38	80.85	94.68	91.16	97.59
SLATT (ours)	42.81	66.56	71.28	93.62	89.16	96.39

Table C.8: Overview single- and multi-shot rank-1 accuracies on OPR (p=32), TVPR (p=94) and DPI-T (p=249) datasets.

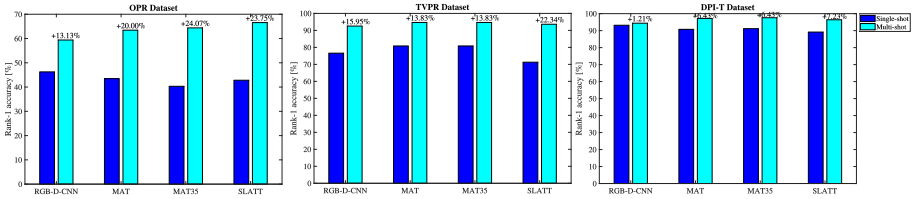


Fig. C.9: Relative increase in rank-1 accuracy from single-shot to multi-shot setting using RGB-D-CNN, MAT, MAT35 and SLATT, respectively, on OPR (p=32), TVPR (p=94), and DPI-T (P=249).

From Figure C.9 it is clear that the addition of the L-ATT module results in an architecture that better captures the individual structures in each image, resulting in an overall larger increase in rank-1 accuracy when fusing features from multiple images. In case of TVPR and DPI-T, the relative increase compared to MAT35 is 8.51% and 0.8%, respectively. Only in the case of OPR do we see similar relative increase when comparing MAT35 and SLATT, however, compared to MAT, the relative increase of SLATT is 3.75% higher.

Interestingly is also the fact that the relative increase of RGB-D-CNN in case of DPI-T and OPR is 1.21% and 13.13%, which is 6.02% and 10.62% worse, respectively, compared to SLATT. This indicates the importance of both capturing local context features using the S-ATT module, and dynamically fuse the features using the L-ATT module.

To further highlight the contribution of the L-ATT module in a setting where identifying the optimal combination of local context features takes much longer, we conduct experiments using a different, deeper, CNN as backbone. We choose an architecture which is comparable to the AlexNet

in terms of complexity, to make the results more comparable to those shown in Figure C.3. Due to its high performance compared to complexity [62], we choose MobileNetV2 [63] as backbone. The network consists of *bottleneck* operators that each consist of up to four identical *bottleneck residual blocks*, where the number of parameters of the layers depend on the bottleneck. The residual blocks each consist of an expansion layer transforming the input from size H×W×C to H×W×kC by 1×1 convolutions, a depthwise 3×3 convolution layer transforming the input from size H×W×kC to H/s×W/s×kC, and a linear layer transforming the output from the depthwise convolution to size H/s×W/s×C′ by 1×1 convolutions. As activation, they use ReLU6, which is a ReLU activation function with an upper bounded value of six. The network consists of seven bottleneck operators, thus, the number of possible combinations of local context features exceeds 5000. Since it is inexpedient to evaluate such a high number of combinations, we compare the result of concatenating the local context features of all seven bottleneck operators to weighting the features using the L-ATT module.

We train RGB and depth CNNs as described in Section 4.1, and afterwards train SLATT and MAT models, respectively. experiments are conducted on TVPR and OPR[2], following the protocols described in Section 4.2. Training and testing the MAT using a 1080 GTX takes ≈4 hours, thus, it would take a long time to find the optimal set of local features using exhaustive search. The experimental results are shown in Figure C.10. On OPR, concatenating features results in a rank-1 accuracy of 70.62%, while the use of dynamic fusion increases rank-1 accuracy by 2.19% to 72.81%. Similarly on TVPR, rank-1 accuracy is increased by 6.38% from a 88.30% to 94.68%. Finally, using MobileNetV2 as backbone in SLATT, rank-1 accuracy is increased by 6.25% and 1.06% on OPR and TVPR, respectively, compared to using AlexNet. From the results it is clear that the proposed SLATT better captures the importance of different local features, while neglecting redundant ones. As a result, only the most informative features are considered, resulting in a higher accuracy.

5 Conclusion

In this work, we combine the use of spatial attention (S-ATT) to capture features at different abstraction levels in a multimodal CNN with dynamic fusion of local context features at different abstraction levels. This is done by introducing a layer-wise attention module (L-ATT), which dynamically weights features based on the input and the multimodal behavior of the entire model. Layer-wise weights are calculated using a soft attention mechanism, which calculates the scores between each of the local context features from the S-

[2]Using RGB alone resulted in a rank-1 accuracy of 99.60% on DPI-T, thus, it does not make sense to do further testing of SLATT or MAT on this dataset.

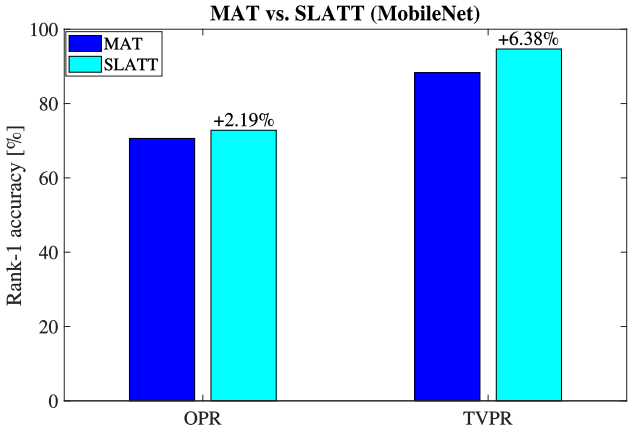


Fig. C.10: Comparison of rank-1 accuracy between concatenation of local features (MAT) and dynamic feature fusion (SLATT) using MobileNetV2 as backbone.

ATT modules and a multimodal reference vector, to determine the relevance of each feature. Thus, a weighted summary of features at all abstraction levels makes up a multi-local feature vector containing local discriminative information. Local and global features are fused in case of both RGB and depth, and a multi-level multimodal feature is finally generated by fusion of modality-based features. Experimental results on two public datasets, DPI-T and TVPR, show rank-1 accuracies of 96.39% and 93.62%, respectively, which are comparable to the existing state-of-the-art systems. Additionally, the state-of-the-art accuracy on a third dataset, OPR, is increased by 3.12% compared to previous work. From a visual analysis of both S-ATT maps and corresponding L-ATT weights, it is shown that the L-ATT module is able to adapt the dynamic weighting to the data. Our results on the datasets OPR and TVPR, which are captured from a more vertical viewpoint, show that head and shoulder regions are highlighted and weighted higher compared to DPI-T. Finally, a quantitative analysis highlights the contribution of the L-ATT module by providing higher relative accuracies when fusing information from multiple images compared to considering a single image. Additionally, using a deeper CNN as backbone, such as MobileNetV2 that consists of several more local context features, dynamic fusion of features results in a higher rank-1 accuracy on both OPR and TVPR.

6 Discussion and Future Work

Based on the experimental results and visual analysis, a clear advantage of the proposed system compared to previous work is its ability to capture useful local context information using the S-ATT, which increases the accuracy as also shown in Table C.2. Additionally, the L-ATT module does not follow a common weighting scheme for all datasets, but adapts to the presented data. Furthermore, it is able to determine the relevance of local context features to the overall multi-modal fusion scheme based on each individual image. This adds a certain robustness to translational and rotational changes. However, challenges arise when the viewpoint becomes more vertical or objects common to the scene are present. The first issue is indicated by the less vertically captured DPI-T dataset, where also more background information is present compared to OPR and TVPR. In this case, discriminative depth information is difficult to exploit since depth maps are more similar across the entire dataset. In this case, a better solution might be to apply a joint localization algorithm, as in [21], to capture relations between body parts. Furthermore, even though, the addition of the L-ATT module show larger minimum, maximum and median rank-1 accuracies on the OPR dataset, more work still needs to be done, to make the method less sensitive to different data distributions in order to minimize variety between tests. In this case, we observed smaller variation in case of MAT. One idea is to also apply dynamic weighting to each frame, as proposed in [37], to suppress, or even neglect, noisy frames.

Acknowledgement

This work is supported by Innovation Fund Denmark under Grant 5189-00222B

References

[1] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person re-identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 1, pp. 1–20.

[2] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, no. 4, pp. 270–286, 2014.

[3] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[4] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[5] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[6] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Proc. CVPR*, 2016, pp. 1363–1372.

[7] Y. Yang, S. Liao, Z. Lei, and S. Z. Li, "Large scale similarity learning using similar pairs for person verification," in *Proc. AAAI*, 2016, pp. 3655–3661.

[8] F. M. Khan and F. Brèmond, "Multi-shot person re-identification using part appearance mixture," in *Proc. WACV*, 2017, pp. 605–614.

[9] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. ICCV*, 2015, pp. 1116–1124.

[10] S. Li, M. Shao, and Y. Fu, "Person re-identification by cross-view multi-level dictionary learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2963–2977, 2018.

[11] K. Li, Z. Ding, S. Li, and Y. Fu, "Toward resolution-invariant person reidentification via projective dictionary learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 6, pp. 1896–1907, 2019.

[12] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[13] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. CVPR*, 2016, pp. 1239–1248.

[14] S. Bai, X. Bai, and Q. Tian, "Scalable person re-identification on supervised smoothed manifold," in *Proc. CVPR*, 2017, pp. 2530–2539.

[15] X. Yang, M. Wang, and D. Tao, "Person re-identification with metric learning using privileged information," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 791–805, 2017.

[16] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. CVPR*, 2015, pp. 3908–3916.

[17] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proc. CVPR*, 2016, pp. 1335–1344.

[18] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-region bilinear convolutional neural networks for person re-identification," in *Proc. AVSS*, 2017, pp. 1–6.

[19] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[20] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, 2014, pp. 34–39.

[21] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *Proc. CVPR*, 2017, pp. 1077–1085.

[22] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proc. ECCV*, 2016, pp. 868–884.

[23] M. Gou, S. Karanam, W. Liu, O. I. Camps, and R. J. Radke, "Dukemtmc4reid: A large-scale multi-camera person re-identification dataset." in *Proc. CVPR Workshops*, 2017, pp. 1425–1434.

[24] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person trasfer gan to bridge domain gap for person re-identification," in *Proc. CVPR*, 2018, pp. 79–88.

[25] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, "Person re-identification dataset with rgb-d camera in a top-view configuration," in *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*. Springer, 2016, pp. 1–11.

[26] M. Saquib Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *Proc. CVPR*, 2018, pp. 420–429.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[28] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "Attention in multimodal neural networks for person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 179–187.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[30] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. CVPR*, 2017, pp. 384–393.

[31] Y. Suh, J. Wang, S. Tang, T. Mei, and K. Mu Lee, "Part-aligned bilinear representations for person re-identification," in *Proc. ECCV*, 2018, pp. 402–419.

[32] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Proc. IROS*, 2015, pp. 681–687.

[33] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. ICRA*, 2015, pp. 1329–1335.

[34] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, and P. Ogunbona, "Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks," in *Proc. CVPR*, 2017, pp. 416–425.

[35] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *Proc. ICRA*, 2017, pp. 4628–4635.

[36] A. Wu, W.-S. Zheng, and J.-H. Lai, "Robust depth-based person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2588–2603, 2017.

[37] N. Karianakis, Z. Liu, Y. Chen, and S. Soatto, "Reinforced temporal attention and split-rate transfer for depth-based person re-identification," in *Proc. ECCV*, 2018, pp. 715–733.

[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015, pp. 2048–2057.

[39] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[40] L. Zhao, X. Li, Y. Zhuang, and J. Wang, "Deeply-learned part-aligned representations for person re-identification," in *Proc. ICCV*, 2017, pp. 3219–3228.

[41] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[42] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proc. CVPR*, 2018, pp. 1179–1188.

[43] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual attention matching network for context-aware feature sequence based person re-identification," in *Proc. CVPR*, 2018, pp. 5363–5372.

[44] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. CVPR*, 2018, pp. 2285–2294.

[45] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Proc. NIPS*, 2015, pp. 2017–2025.

[46] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. CVPR*, 2017, pp. 3156–3164.

[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[48] A. Haque, A. Alahi, and L. Fei-Fei, "Recurrent attention models for depth-based person identification," in *Proc. CVPR*, 2016, pp. 1229–1238.

[49] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proc. CVPR*, 2018, pp. 369–378.

[50] X. Zhang, K. Gao, Y. Zhang, D. Zhang, J. Li, and Q. Tian, "Task-driven dynamic fusion: Reducing ambiguity in video description," in *Proc. CVPR*, 2017, pp. 3713–3721.

[51] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in *Proc. ICCV*, 2017, pp. 4203–4212.

[52] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen, "Multimodal keyless attention fusion for video classification," in *Proc. AAAI*. 7202–7209, 2018.

[53] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proc. CVPR*, 2017, pp. 1831–1840.

[54] H. Chen, Y.-F. Li, and D. Su, "Attention-aware cross-modal cross-level fusion network for rgb-d salient object detection," in *Proc. IROS*, 2018, pp. 6821–6826.

[55] X. Chang, T. M. Hospedales, and T. Xiang, "Multi-level factorisation net for person re-identification," in *Proc. CVPR*, 2018, pp. 2109–2118.

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

[57] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.

[60] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. CVPR*, 2017, pp. 7263–7271.

[61] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[62] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[63] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

Paper D

Enhancing Person Re-identification by Late Fusion of Low-, Mid-, and High-Level Features

Aske R. Lejbølle, Kamal Nasrollahi, and Thomas B. Moeslund

The paper has been published in the
IET Biometrics Vol. 7(2), pp. 125–135, 2018.

Please note that this article is based on the following Master's thesis: Aske R. Lejbølle, *"Enhancing Person Re-identification by Late Fusion"*, Master's thesis, Aalborg University, Denmark, 2016. Thus, overlap occurs in tables and figures between the thesis and this article. The Master's thesis has been added as a reference [1], and citations are added wherever necessary to make any overlap evident.

The layout has been revised.

Abstract

Person re-identification is the process of finding people across different cameras. In this process, focus often lies in developing strong feature descriptors or a robust metric learning algorithm. While the two aspects are the most important steps in order to secure a high performance, a less explored aspect is late fusion of complementary features. For this purpose, this paper proposes a late fusing scheme that, based on an experimental analysis, combines three systems that focus on extracting features and provide supervised learning on different abstraction levels. To analyse the behaviour of the proposed system, both rank aggregation and score-level fusion are applied. Our proposed fusion scheme increases results on both small and large datasets. Experimental results on VIPeR show accuracies 5.43% higher than related systems, while results on PRID450S and CUHK01 increase state-of-the-art results by 10.94% and 14.84%, respectively. Furthermore, a cross dataset test show an increased rank-1 accuracy of 28.26% when training on CUHK02 and testing on VIPeR. Finally, an analysis of the late fusion shows aggregation to be better when individual results are unequally distributed within top-10 while score-level fusion provides better results when two individual results lie within top-5 while the last lies outside top-10.

1 Introduction

Person re-identification is of great importance in biometrics and surveillance systems [2–4]. It is defined as the task of comparing images of persons captured by cameras with different views for the purpose of finding matches. Given an image (probe) from camera A, it is compared against all other images (gallery) captured by camera B. The results are ranked according to an employed similarity measure to find the most similar images in the gallery for the given probe. Since the viewpoint of each camera is different along with the environment in which they are placed, problems like, viewpoint variations, lighting, scale, pose, and occlusion make re-identification a very challenging research topic (see Figure D.1).

When dealing with re-identification, two steps are very important in order to reach good performance: 1) feature extraction and representation and 2) metric learning. Regarding the first step, feature extraction and representation, the employed feature descriptors should not only be discriminative but also fast to extract as they are compared against a potentially large database [5, 6]. Features can either be extracted globally by looking at larger areas such as body parts [7] or locally by sampling minor local patches [8]. Furthermore, features can be extracted on different abstraction levels by dividing into low-, mid- and high-level depending on representation of the features. Low-level features are per-pixel based and typically cover colour and texture histograms. Examples of more advanced low-level features in-

Fig. D.1: Examples of image pairs from different datasets, including; (a) VIPeR, (b) PRID450S, (c) CUHK01 and (d) CUHK03.

clude creating a covariance matrix from image derivatives [9] or looking at local key points as in scale-invariant feature transform (SIFT) [10]. Mid- and high-level features are both learned from low-level features and are defined by representing either parts of objects in the image (mid) or the entire object in the image (high) by sparse feature representations. Examples of mid-level features are bag of words (BoW) models [11] used to quantize low-level features to visual words while the output from one of the late layers in a convolution neural network (CNN) is an example of a high-level feature representation [12].

Regarding the second step, metric learning, supervised learning is often utilized in which a selection of image pairs from a number of persons are used to compute a distance matrix used when calculating the similarities [13] or compute a projection matrix used to project features to a subspace [14]. Common for all methods is emphasizing on keeping features from dissimilar pairs apart while keeping features from similar pairs as close as possible.

In order to enhance the performance of a re-identification system, fusion can be applied on different levels. In the multi-shot case i.e., when several images of each identity are available, data-level fusion can be applied by combining information from all images of the same identity. To have a more robust representation of each identity, fusion on feature level can be applied instead. Feature fusion is typically applied by fusing different types of colour and texture features, often by simple concatenation. Fusing of different feature types makes sense if the features do not result in redundant information and complement each other. Dimensionality reduction techniques, such as principal component analysis (PCA), are therefore often used to reduce such fused features. Finally, fusion can be applied on late level [15] as well. In this case, the outputs of different re-identification systems are fused, usually

using Bayesian decision theory to combine the outputs by either summing or multiplying the scores [16]. In such a situation, a weight can be assigned to each of the scores depending on the correctness computed from training data. Another way is to combine the ranked lists by aggregation, either by looking at the mean position of each ranked identity or using order statistics [17].

In this paper, we analyse the complementarity of features extracted at different abstraction levels by applying late fusion on different combinations of outputs from three re-identification systems, each working at different abstraction levels. This ends up in a system that advances state-of-the-art re-identification results on public benchmark datasets. For a more extensive analysis, we apply two different well known late fusion techniques, score-level fusion based on Bayesian theory [16] and rank aggregation [17]. We analyse the scenarios in which late fusion improves the results in order to define situations in which either of the late fusion techniques provides better results. This include an analysis of how different information captured by different feature types affect the late fused result. Finally, we measure processing time of late fusion and compare it with the increased accuracy. The contributions of this paper are therefore as follows:

1. We show that fusion of low-, mid- and high-level features is of potential when late fusion is applied.
2. We analyse how different feature types positively affect late fusion.
3. We conclude the cases in which it is better to apply score-level fusion than rank aggregation and vice versa.
4. We show that late fusion does not add particular processing time compared to increased accuracy.

The rest of this paper is organized as follows: first, the related work within re-identification is reviewed in Section D.2. Then, the proposed system, including the fusion techniques and the three chosen re-identification features which are used in the fusion are explained in Section D.3. In Section D.4, experimental results on different public benchmark datasets along with an analysis and measured processing times are reported. Finally, the paper is concluded in Section D.5.

2 Related Work

Low-level features In current re-identification systems, different low-level features are typically fused in order to take both colour and texture information into account. In order to make the features more discriminative, features are extracted more locally as those by Zhao *et al.* [10], that are obtained

by sampling of local patches from which SIFT descriptors and colour histograms are extracted. The patches are then used to learn a set of support vector machines (SVMs) by clustering similar patches. Local patches are also used in [18], in which a Gaussian function is used to calculate a similarity score along with k-nearest neighbors (KNN) to find the most similar reference patches for each test patch in an unsupervised manner. Finally, Liao *et al.* [19] extract colour and texture histograms from overlapping patches and use a metric learning algorithm based on Mahalanobis distance to learn a projection matrix used to keep distance between similar image pairs closer. An example of low-level features that are extracted more globally is given in [20], in which the body is horizontally split into six equally sized stripes and colour name features are extracted and used with KISSME [13] metric learning. Similar regions are used in [7, 21, 22], in which different colour and texture based features are extracted and used for cross-view metric learning.

Mid-level features Few systems utilize low-level features to learn mid-level features. An example of this is the BoW model learned in [23], which is used to extract mid-level features that are used together with a cosine similarity in an unsupervised manner. Another example of mid-level features is given in [24], in which dictionaries based on hue, stauration, value (HSV) colour and local binary patterns (LBP) texture histograms are learned to represent features as *atoms* that are contained in the dictionary.

High-level features As the popularity of CNN increases, they have also been proposed for person re-identification as they are able to learn high-level feature representations by training on images without the need for hand-crafted low- or mid-level features. Usually, Siamese networks are constructed, taking an image pair as input and outputting whether they match or not [5, 25, 26]. Another idea has been proposed in [27], in which a pre-trained CNN extracts high-level features used along with hand-crafted low-level features. Finally, [28] extracts hand-crafted low-level features that are included in training a CNN to make a more robust image representation. High-level features are then extracted using the trained CNN and applied to a metric learning algorithm.

Late fusion In the context of re-identification, only a few systems propose late fusion of results to increase the overall performance. In [29], the product rule from Bayesian probability theory is used to fuse scores calculated as the dot product between two feature vectors. Each feature is furthermore assigned a weight which is calculated from the area under the feature's score curve. Score-level fusion is also applied in [30], in which the outputs of utilizing different metric learning algorithms are fused. In this case, they are all trained using similar low-level features. A different way of combining scores was proposed in [31], in which linear combinations of scores are computed using a weight, which is learned through SVM. Rank aggregation was used in [32], in which ranking lists, calculated using both locally and more globally

extracted low-level features, are combined. Aggregation is also used in [33] where different ranking lists computed using both individual and concatenated low-level features are aggregated.

3 Proposed System

The block diagram of the proposed system is shown in Figure D.2. First, features on different abstraction levels are extracted from a given probe and transformed using the trained metric learning algorithms. Next, similarities are calculated between probe and gallery features in each of the learned subspaces. The resulting outputs are then either fused using the scores or ranks of the identities. The output is a new ranked list which is re-ordered. These steps are explained in the following sub-sections, along with the used features.

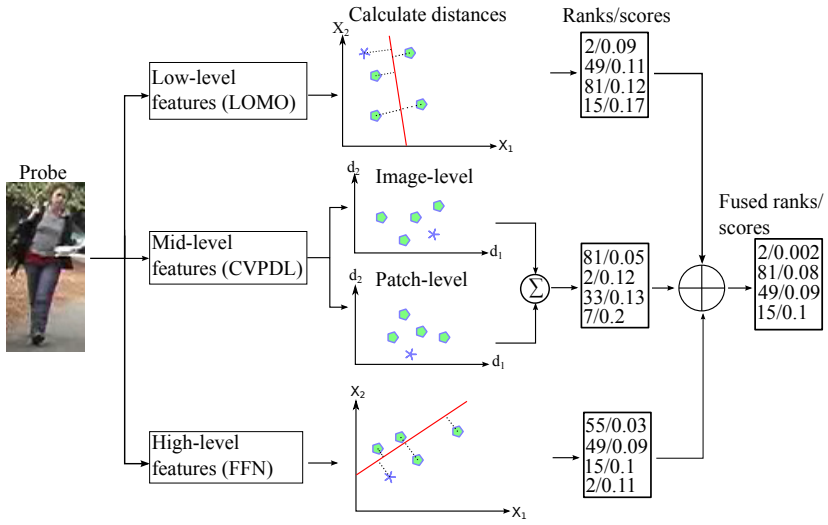


Fig. D.2: Overview of fusion system. Given a probe, low-, mid- and high-level features are first extracted. Then, extracted features (star) are transformed either to a subspace (LOMO and FFN) given by the solid red lines or changed to a different representation (CVPDL) and matched with the gallery (polygons). For simplicity, only features in two dimensions are shown. The outputs are then fused by a specified late fusion technique (symbolised by $\oplus$), producing a fused output.

3.1 Low-level features

The first part of the fusing system emphasizes on low-level features extracted from local patches to make the features more discriminative. A benchmark is provided by Schwarz *et al.* which compares results from different systems

tested on different datasets [Available at http://www.ssig.dcc.ufmg.br/reid-results/]. By a review of the systems that show the best results, the system proposed by Liao *et al.* [19] is chosen as it not only achieves decent results on most of the listed datasets with single-shot rank-1 accuracies of 40% and 52.20% on VIPeR and CUHK03, respectively, but also shows fast feature extraction and metric learning.

The system, as shown in Figure D.3, works by preprocessing each image using the Retinex algorithm [34], to enhance colour information, especially in shadowed regions. Next, features of overlapping patches of size 10×10 are extracted, including a joint HSV histogram with a bin size of 8, along with Scale Invariant Local Ternary Patterns (SILTP) [35] to handle illumination changes. For patches located at same horizontal level, histograms within each channel are compared and maximized to deal with viewpoint changes between camera views. Furthermore, the image is downsampled two times by applying a 2×2 average pooling kernel and a similar feature extracting procedure is carried out. Finally, a log transformation is applied to each histogram to suppress large bin values. Due to feature maximization, the method is called Local Maximal Occurrence Representation (LOMO).

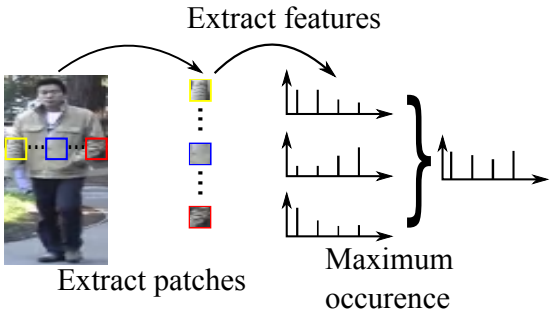


Fig. D.3: Low-level features from LOcal Maximal Occurrence Representation (LOMO) that are composed of colour and texture features [19].

Following [19], an extended version of KISSME, shown in Equation D.1, is used as metric learning. This algorithm is originally based on Mahalanobis distance:

$$d_M^2(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j), \tag{D.1}$$

where the distance matrix $M = \Sigma_S^{-1} - \Sigma_D^{-1}$ with Σ_S calculated from similar image pairs and Σ_D from dissimilar pairs. These two matrices represent the intra-class and inter-class differences.

The metric learning is called Cross-view Quadratic Discriminant Analysis (XQDA) and considers a projection matrix, W, which maps features from two different views to a subspace before calculating the distance, as given by Equation D.2:

$$d_M^2(x_i, x_j) = (x_i - x_j)^T W M' W^T (x_i - x_j), \tag{D.2}$$

where $M' = \Sigma_S'^{-1} - \Sigma_D'^{-1}$ and the two matrices, Σ_S' and Σ_D', are computed as shown in Equation D.3:

$$\Sigma_S' = W^T \Sigma_S W \quad \Sigma_D' = W^T \Sigma_D W \tag{D.3}$$

The matrix W is computed by maximizing the ratio of the intra-class and inter-class variance. As this can be calculated from an eigenvalue decomposition of the two matrices in M, training time is only 0.5 seconds slower than KISSME.

3.2 Mid-level features

The second part of the fusing system utilizes mid-level features. From the few proposed systems that make use of sparse coding, the features developed by Li *et al.* [36] are chosen as the method utilizes dictionary learning from patches on both patch- and image-level. Furthermore, the system achieves decent results with rank-1 accuracies of 33.99% and 59.47% on VIPeR and CUHK01, respectively.

It extracts features in Lab colour space from overlapping patches of size 10×10 with a step size of 5. For each patch, 32-dimensional colour histograms and 128-dimensional SIFT descriptors are extracted in each channel. In addition, colour features are extracted from down sampled patches using scaling factors of 0.5 and 0.75, resulting in a total 672-dimensional feature vector for each patch. Finally, all features are L_2 normalized before the training set is used for dictionary learning. The mid-level test features are extracted by utilizing the learned dictionaries along with Orthogonal Matching Pursuit to transform from low-level features as shown in Figure D.4. While features are transformed for each patch at patch-level, all patch features are concatenated and transformed on image-level.

Instead of solving the usual objection function when dealing with dictionary learning, defined by
$\min_{D,Z} \|X - DZ\|_F^2$ s.t. $\|d_i \leq 1\|_2$, where X is the feature matrix, D is the dictionary and Z is the coefficient matrix, Li *et al.* [36] propose a projection matrix, P, in order to ease the NP-hard problem. Given features X, dictionaries D and projection matrices P in camera views *1* and *2*, the objection function defined in Equation D.4 is solved as:

$$\begin{aligned} \min_{D_1, D_2, P_1, P_2} \; & ||X_1 - D_1 P_1 X_1||_F^2 + \\ & ||X_2 - D_2 P_2 X_2||_F^2 + \lambda f(D_1, D_2, P_1, P_2) \\ \text{s.t. } & ||d_{1,i} \leq 1||_2, \quad ||d_{2,i} \leq 1||_2, \end{aligned} \tag{D.4}$$

where $f(D_1, D_2, P_1, P_2)$ is a regularization function which affects the similarity between dictionary or projection matrices in the two views. As dictionary and projection matrices are learned on both patch- and image-level, the superscripts L and H are used to represent patch-level and image-level, respectively.

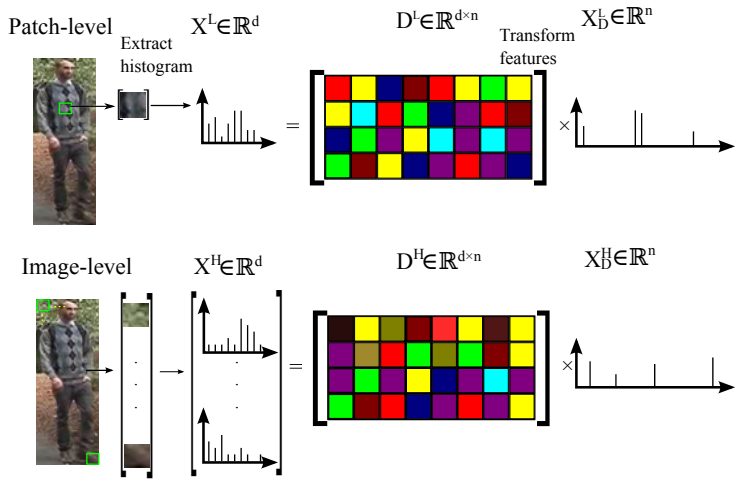


Fig. D.4: Mid-level features by Cross-View Projective Dictionary Learning (CVPDL) [36]. Each patch feature is transformed using a learned dictionary at patch-level while all patch features are concatenated and transformed on image-level.

At patch-level, patches at the same spatial location are assumed to share the same dictionary, making the regularization function $\|D_1^L - D_2^L\|_F^2$ and by splitting each of the first two terms in Equation D.4 by adding a relaxation variable, A, the dictionary and projection matrices are calculated by solving the objection function defined in Equation D.5:

$$\begin{aligned} \min_{D_1^L, P_1^L, A_1^L} \; & ||X_1 - D_1^L A_1^L||_F^2 + \beta ||P_1^L X_1 - A_1^L||_F^2 + \\ & \lambda_1 ||D_1^L - D_2^L||_F^2 \\ \text{s.t. } & ||d_{1,i}^L \leq 1||_2, \quad ||d_{2,i}^L \leq 1||_2, \end{aligned} \tag{D.5}$$

where β is a balance parameter. Similar objection function is used to calculate D_2^L, A_2^L and P_2^L.

At image-level, all patch features are concatenated to a single feature representation and features in the two views are instead assumed to share the same subspace i.e., projection matrix, resulting in a regularization function $\|P_1^H X_1 - P_2^H X_2\|$. By once again introducing the relaxation variable, the ob-

jection function on image-level in one view is defined by Equation D.6:

$$\begin{aligned} \min_{D_1^H, P_1^H, A_1^H} ||X_1 - D_1^H A_1^H||_F^2 + \alpha||P_1^H X_1 - A_1^H||_F^2 + \\ \lambda_2||A_1^H - A_2^H||_F^2 \\ \text{s.t. } ||d_{1,i}^H \leq 1||_2, \quad ||d_{2,i}^H \leq 1||_2, \end{aligned} \tag{D.6}$$

As in the case of patch-level, a similar objection function is defined to calculate D_2^H, P_2^H and A_2^H.

When matching at patch-level, each patch in view 1 is compared with every patch at same horizontal level in view 2 to account for misalignment. The shortest distance is then defined as the distance for that particular patch. Having calculated distances for all patches, the scores are accumulated to determine the score, $score_P$, between a probe in view 1 and a query in view 2. At image-level, the score between a probe and a query, $score_I$, is calculated as the cosine similarity.

Finally, the patch- and image-level scores are fused by $Score = score_P + \lambda score_I$, where λ is a pre-defined weight parameter between [0,1].

3.3 High-level features

The third and final part of the fusing system emphasizes on high-level features. As CNN's are both fast and have shown decent results, utilizing such a network is desired. Most of the already proposed CNN's produce a binary output to whether an image pair match or not. This is not suitable in late fusion as the probability often will be large for either being similar or dissimilar and the CNN will therefore overrule decisions made by other systems. Wu *et al.* [28] propose a system which combines CNN and low-level hand-crafted features in a new architecture called feature fusion network (FFN). This way, a new type of feature is learned by both taking low-level colour and texture features along with more high-level CNN features into account. The system achieves rank-1 accuracies of 41.69%, 47.53% and 58.02% on VIPeR, CUHK01 and PRID450s, respectively. Furthermore, the new feature type is fused with LOMO features on feature level, increasing rank-1 accuracies to 51.06%, 55.51% and 66.62%, respectively.

The architecture, as shown in Figure D.5, consists of two parts, the top being a CNN and the bottom being extraction of hand-crafted features.

The CNN architecture is similar to the AlexNet architecture presented by Krizhevsky *et al.* [12] and consists of five convolution layers, all but the third followed by a MAX pooling and normalization layer. The network takes a randomly cropped image of size $227 \times 227 \times 3$ as input and outputs a 4096-dimensional CNN feature vector from the last pooling layer.

The other part horizontally divides the input image to 18 equally sized images and extracts colour histograms using colour spaces RGB, HSV, Lab,

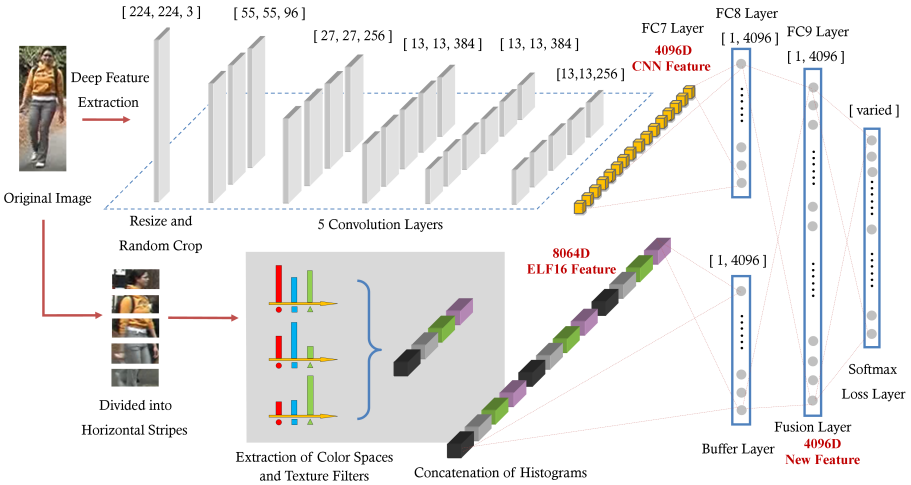


Fig. D.5: High-level features from Feature Fusion Network (FFN) [28]. Top part outputs CNN features from a pre-trained CNN while the bottom part outputs hand-crafted features. The second fully connected layer (FC9) outputs the new FFN feature. The figure is from [28].

YCbCr and YIQ along with Gabor texture features. All histograms are represented in 16 bins and are L_1-normalized before concatenated to a 8064-dimensional feature vector.

Having two different feature vectors, they are each fed to fully connected layers called **buffer** layers that output two 4096-dimensional feature vectors. The two outputs are then concatenated and fed to the new **fusion** layer which weights are learned based on the feature types. Connecting the two features types, Wu *et al.* [28] show that the weight update for the CNN is influenced by the output of the hand-crafted features. The output from the fusion layer is then defined as the new FFN feature type. In the training phase, a softmax layer is used to determine the corresponding label based on the input.

The FFN is trained using a fine-tuning scheme in which a pre-trained AlexNet model is utilized. The network is fine-tuned for 50,000 iterations using Stochastic Gradient Descent (SGD) with a mini-batch size of 25. After fine-tuning, FFN features are extracted from the fusion layer. As shown in the bottom of Figure D.2, a projection matrix is then learned to map features to a subspace. In this case, mirror kernel marginal fisher analysis (mirror-KMFA) training scheme proposed by Chen *et al.* [37], with a chi-square kernel, is used as metric learning algorithm because of its high performance. After converting features to kernel space, the algorithm aligns the feature distributions from features in two different views and calculates mirror transformed

features using a projection matrix as defined in Equation D.7:

$$X = \Lambda^{-\frac{1}{2}} U^T X^k_{aug}, \tag{D.7}$$

where X^k_{aug} is the kernalized and augmented feature matrix while Λ and U are matrices containing eigenvalues and corresponding eigenvectors, respectively, from an eigenvalue decomposition of a matrix $C = [K, -\beta K; -\beta K, K]$ with K being a matrix containing features from both views while β is a regularization term.

Next, marginal fisher analysis (MFA) is utilized by computing a projection matrix M, by solving the generalized eigenvalue problem $S^w m_i = \lambda_i S^b m_i$ where S^w and S^b are intra- and inter-class scatter matrices, respectively. Finally, similarity between features is calculated in kernel space using Equation D.8:

$$\mathcal{D}_{x_{1,i}, x_{2,j}} = x^2_{1,i} + x^2_{2,j} + 2x_{1,i}x_{2,j} \tag{D.8}$$

3.4 The proposed late fusion

Two different late fusion techniques are used in this paper, as they both have shown decent results when applied in other contexts.

Score-level fusion The first late fusion method is based on the computed similarity scores, for each method. The algorithm was proposed by Zheng *et al.* [29] and is based on Bayesian theory of combining classifiers. In this case, the product rule is used since it has shown to be superior to other ways of combining outputs [16].

Having a number of computed similarity scores, $s^{(i)}_{p,q}$, where i denotes the method and p, q denotes a probe and a query image, the late fused output is calculated following Equation D.9:

$$sim(p, q) = \prod_{i=1}^{K} (s^{(i)}_{p,q})^{w^{(i)}_q}, \qquad \sum_{i=1}^{K} w^{(i)}_q = 1, \tag{D.9}$$

where $w^{(i)}_q$ is the weight assigned for the i^{th} method and originally calculated using Equation D.10:

$$w^{(i)}_q = \frac{\frac{1}{AUC_i}}{\sum_{k=1}^{K} \frac{1}{AUC_k}}, \tag{D.10}$$

where AUC_i is the area under the score curve (AUC) for the i^{th} method.

In the original context, higher scores are better and they therefore find an equal reference score, calculated from the training samples, used to remove

the tail of the score curve and decrease the AUC. This is done by calculating the euclidean distance between a predefined number of reference curves and the test curve, The k-nearest reference curves are then averaged and subtracted the test curve.

Although, in our case, a lower score value is better, leaving us with another way to deal with the reference curves and weight assignment, yet following same procedure by finding the most similar reference curves to the test score curves. An example of a score curve and the averaged nearest reference curves is shown in Figure D.6 along with the resulting curve after subtraction.

As we desire to keep the AUC large by having small scores for the most similar pairs and large for the rest, we flip the reference curve before subtraction. This way, we keep the score for most similar pairs to a minimum while increasing the total AUC as shown in Figure D.6 (b).

The weight assignment is, hence, also changed to be calculated following Equation D.11:

$$w_q^{(i)} = \frac{AUC_i}{\sum_{k=1}^{K} AUC_k} \tag{D.11}$$

Lastly, in order to properly make use of this method, a common metric for all outputs is needed to avoid any bias. Therefore, the scores are min-max normalized by Equation D.12, before they are fused:

$$\hat{d}_M^2(x_i, x_j) = \frac{d_M^2(x_i - x_j) - \min d_M^2(\mathbf{x})}{\max d_M^2(\mathbf{x}) - \min d_M^2(\mathbf{x})} \tag{D.12}$$

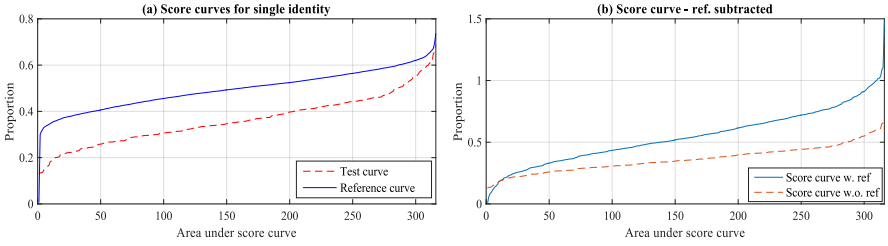


Fig. D.6: *The impact of subtracting a reference curve: (a) Original test score curve along with the average of the most similar reference score curves, (b) New (Score curve w. ref) test score curve obtained by flipping and subtracting averaged reference score curve compared to Original (Score curve w.o. ref).* [1]

Rank aggregation Instead of looking at scores, rank aggregation makes use of the ranking lists for each method. The technique was successfully used by Ye *et al.* [32] who combine ranking lists from locally and globally based features using KNN to increase accuracy. Some of the most common ways of

re-ranking is either by looking at the median or mean ranking or looking at the maximum ranking among all ranking lists.

In [38], different ranking aggregation techniques are compared, including Mean, Max, Stuart [39] and Robust Ranking Aggregation (RRA) [40]. It is shown that the Stuart method is slightly better than the others. The technique was originally used for aggregation of lists of genes which would contain noisy information and is therefore made robust to this challenge. It is therefore suited for person re-identification and is used in this context.

The technique take into account the statistically placement of the ranks, $r_{(i)} = \{r_{i,1}, ..., r_{i,m}, ..., r_{i,M}\}$ to calculate a new order of the ranks $r_{(i)}^{new}$ where $r_{i,m}$ is the rank for the i^{th} person at the m^{th} output, following Equation D.13:

$$r_{(i)}^{new} = M! \cdot V_{M+1}, \tag{D.13}$$

with $V_{M+1} = \sum_{m=1}^{M} \sum_{l=1}^{m} (-1)^{l-1} \frac{V_{m-l}}{l!} r_{M-m+1}^{l}$ and $V_0 = 1$.

Using this formula, ranks are first normalized to the interval [0,1] where smaller values indicate a higher rank.

4 Experimental results

In this section we first give the details of the datasets and protocols that are used for evaluation of the proposed system. Then, the obtained results from the experiments using different combinations of feature types are analysed. The best results are compared against state-of-the-art re-identification systems and, finally, the processing time of the system is compared to the increase in accuracy. Throughout all tests, we refer to our results from rank aggregation and score-level fusion by subscripts *agg* and *sco*, respectively.

4.1 Datasets and Protocol

Datasets Tests are conducted on four different public datasets, two minor and two larger. Common for all is the challenges of change in viewpoint and illumination which make the task more difficult. Examples of image pairs in each dataset are shown in Figure D.1. Further, a cross-dataset test is conducted for a more realistic performance evaluation. The first dataset is VIPeR [7] which consists of 1264 images of 632 identities captured in two different camera views, thus, each person has one image from each view. The second dataset, PRID450S [41] contains 900 images of 450 identities.

The two larger datasets, CUHK01 and CUHK03 [5, 42], contain multiple images of each person in each camera view. CUHK01 contains 3884 images of 971 different identities. For each identity, two images are captured in each camera view. CUHK03 contains 13,164 images of 1360 different identities.

Images are captured using three camera pairs and images from one identity are captured by one camera pair. One to five images are captured from each identity in each camera view with an average of 4.8. Both manually labeled bounding boxes and automatically detected are included.

Evaluation protocols In all experiments, single-shot setting is used. For VIPeR and PRID450S, identities are randomly divided in a training set of 316 and 225 identities, respectively, while the other half is used for testing. For CUHK01, 485 identities are randomly used for training while 486 are used for testing. For each identity, one image in each camera view is randomly chosen in each iteration. For CUHK03, the protocol defined in [5] is used, having 1160 identities for training and 100 for testing. As for CUHK01, one image in each camera view is randomly chosen in each iteration. For VIPeR, PRID450S and CUHK01, 10 iterations are run, each with randomly split data. For CUHK03, 20 iterations are run following the protocol. The tests on this dataset is made on the manually labelled bounding boxes. The mean accuracy over all iterations is calculated for each dataset and the results are presented by Cumulated Matching Characteristic (CMC) curves that show the accumulated ranked similarities for all identities, having the rank-1 accuracy indicating the number of probes that have their corresponding gallery image as the most similar.

4.2 The results of late fusion

Initial tests are conducted by evaluating different combination of features when late fusion is applied, to conclude which combinations benefit mostly from late fusion. To consider both minor and large datasets, tests are conducted on CUHK01 and VIPeR using the protocols defined in Section D.4. Tables D.1-D.2 summarize the results. All_{agg} and All_{sco} indicate late fusion of all three systems while the results from LOMO, FFN and CVPDL are reproduced and therefore differs from original.

In the case of VIPeR, the best pairwise combination is FFN+LOMO_{agg} with a rank-1 accuracy which is 3.89% higher than the next best result of LOMO+CVPDL_{sco}. Though, when rank aggregation is applied using all three systems, the rank-1 accuracy increase by 1.77% compared to FFN+LOMO_{agg}.

For CUHK01, the best pairwise combination is LOMO+CVPDL_{agg}, while late fusion of all systems achieves a rank-1 accuracy which in comparison is 3.29% higher when rank aggregation is utilized. From this it can be concluded that the best results are achieved when applying late fusion to all three systems while, generally, rank aggregation is shown to provide the best results. Only in the case of combining FFN and LOMO, accuracies are similar for both type of late fusion.

Table D.1: *Results on VIPeR (p=316). Best results are in bolt.* [1]

System/Rank	r = 1	r = 5	r = 10
All_{agg}	**45.63**	**75.06**	85.16
All_{sco}	45.24	75.02	**85.70**
FFN+$LOMO_{agg}$	43.86	73.89	84.34
FFN+$LOMO_{sco}$	43.73	73.81	84.08
FFN+$CVPDL_{agg}$	39.40	68.04	79.59
FFN+$CVPDL_{sco}$	37.85	67.56	78.96
LOMO+$CVPDL_{agg}$	37.22	69.84	81.61
LOMO+$CVPDL_{sco}$	39.94	71.06	83.01
LOMO	37.72	67.59	80.06
FFN	30.70	57.72	69.15
CVPDL	28.23	55.41	70.54

Table D.2: *Results on CUHK01 (p=486). Best results are in bolt.* [1]

System/Rank	r = 1	r = 5	r = 10
All_{agg}	**70.35**	**88.46**	**93.29**
All_{sco}	64.67	83.81	89.35
FFN+$LOMO_{agg}$	47.35	73.19	80.88
FFN+$LOMO_{sco}$	47.17	71.39	78.99
FFN+$CVPDL_{agg}$	65.86	86.85	93.02
FFN+$CVPDL_{sco}$	58.60	80.11	86.78
LOMO+$CVPDL_{agg}$	67.06	87.16	91.89
LOMO+$CVPDL_{sco}$	63.86	84.42	89.74
LOMO	41.77	66.19	74.86
FFN	32.28	56.95	66.73
CVPDL	53.44	78.85	86.95

For both datasets, it is clear that individual results are important for late fusion, although, there is an indication that mid- and high-level features complement each other better than other combinations. For VIPeR, FFN+$CVPDL_{agg}$ result in the highest increase in accuracy of 8.7% compared to individual results of FFN and CVPDL, while increasing rank-1 accuracy by 12.42% in the case of CUHK01. While LOMO+$CVPDL_{agg}$ show the largest increase of 13.62% in the case of CUHK01, there is a decrease of 0.50% in the case of VIPeR.

Comparisons between the late fused system and individual results are furthermore visualized in Figure D.7.

For VIPeR, PRID450S and CUHK03, LOMO provides the best individual results while CVPDL show the highest accuracy on CUHK01 of 53.44%.

While FFN and CVPDL show close to similar results on both VIPeR and PRID450S, CVPDL seem to be better at handling change in texture across camera views as it is superior to FFN in the cases of CUHK01 and CUHK03. When applying score-level fusion, rank-1 accuracies of 45.24%, 68.76%, 64.67 and 54.72% on VIPeR, PRID450S, CUHK01 and CUHK03, respectively, are achieved. Meanwhile, when rank aggregation is applied, rank- accuracies of 45.63%, 77.56%, 70.35% and 52.25% for VIPeR, PRID450S, CUHK01 and CUHK03, respectively, are achieved.

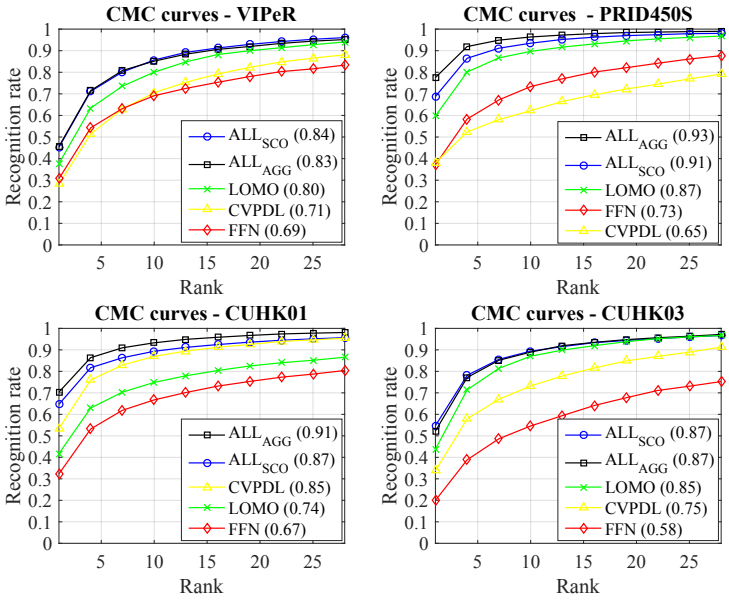


Fig. D.7: CMC curves of results by applying late fusion on (a) VIPeR (p=316); (b) PRID450S (p=225); (c) CUHK01 (p=486) and (d) CUHK03, manually labeled (p=100).

For VIPeR, PRID450S and CUHK01, the largest improvement is achieved by utilizing rank aggregation when compared to the individual results. In these cases, the rank-1 accuracies are increased by 7.91%, 17.83% and 16.91%, respectively. In case of CUHK03, score-level fusion provides the best rank-1 accuracy which increases the results of LOMO by 11.51%.

Comparing score-level fusion and rank aggregation, the latter achieves a rank-1 accuracy which is 0.39% higher in case of VIPeR. For PRID450S and CUHK01, differences are more significant with rank aggregation achieving rank-1 accuracies that are 8.8% and 5.68% higher than score-level fusion. Finally, score-level fusion achieves an accuracy 2.47% higher than rank aggregation in case of CUHK03. This is most likely due to the much worse result by CVPDL, showing score-level fusion to be more robust to single bad performing features.

Overall, large increase in accuracies are achieved on both minor and larger datasets, showing the benefit of applying late fusion to features at different abstraction levels. In addition, the system of [30] increases the rank-1 accuracy by 5.82% compared to their individual results, while we show an increased accuracy of 7.91%.

4.3 The importance of late fusion

To analyse the affection of late fusion, three examples are provided, showing how the fused result is improved or maintained by having different individual results. For each example, an analysis of how different feature types contribute to the late fusion, and how this affect the result, is conducted.

Examples are created by looking at the results from a single test iteration on VIPeR dataset.

The first example is shown in Figure D.8 where all individual systems rank the true match as the fourth most similar, while the remaining queries are ranked differently by each system. For FFN, the contours of the jeans and dark shirt seem to dominate the matched queries, while the results of LOMO and CVPDL show a higher dependency on the colours, especially seen by the impact of change in colour of the jeans. Furthermore, the two latter also seem to be more affected by textures created from shadows present in the probe image. Due to these differences, the aggregated output provides a better result. For the case of score-level fusion, the calculated distances for each system affect the fused result causing the score-fused to be similar.

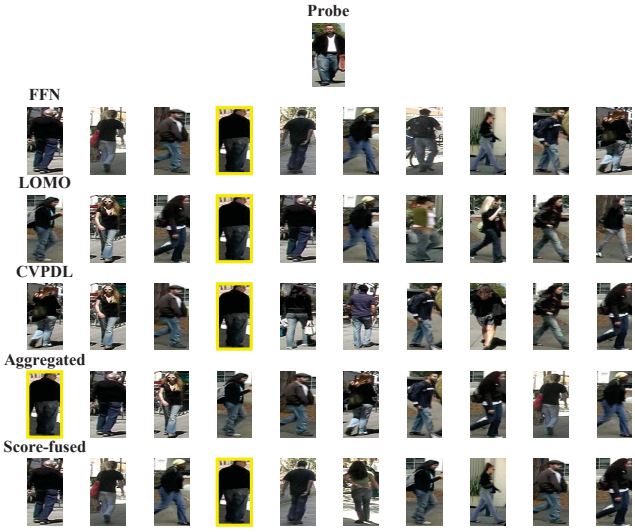


Fig. D.8: First case in which all individual features rank the true match similar. True match is shown by the green rectangle.

In the second example shown in Figure D.9, neither of the individual systems ranks the true match within top-5 while remaining rankings differ even more than in the first example. Here, FFN not only captures information on the contours, but also the colours, since hand-crafted colour features were also used to train FFN. Again, LOMO and CVPDL seem to capture information from the street, such as shadows. But while CVPDL seem to capture more information on the lower part of the image, including the colours of the jeans and texture of the street, LOMO emphasizes more on the combination of colours such as the colour of the left handbag and coat. After applying late fusion, the results are improved in both cases and the output of the rank aggregation now ranks the true match within top-4.

Fig. D.9: Second case in which the true match is ranked differently by each individual feature. True match is shown by the green rectangle.

In the third and last example shown in Figure D.10, LOMO and FFN rank the true match within top-5. While LOMO is affected by the gray-scale change of the trousers due to lighting changes, FFN emphasizes more on the broad body contour, making it more robust to such challenges. Once again, CVPDL seem to capture the texture from shadows, while also capturing the white colours of the shirt. In this case, the rank aggregation is affected by the bad result from CVPDL and the aggregated result is therefore similar to that of LOMO. Meanwhile, score-level fusion is not similarly affected due to the weight assignment and use of distances and the result is therefore similar to that of FFN.

Generally, aggregation performs better than score-level fusion when all three individual systems rank the true match within top-10 while score-level

Fig. D.10: Third case in which the true match is ranked high by two features and outside top-10 by the last. True match is shown by the green rectangle.

fusion achieves better results in situations where one system performs badly, though, this cannot be completely defined as the rankings of other identities affects the fused result. FFN takes the overall contours into account and is therefore not affected much by the background. Contrary, as colour features on a semi-global scale are also used for training, it is affected by changes in colour as seen in Figure D.9. Meanwhile, LOMO performs better when similar colours are preserved in parts of the image, due to utilization of patches. Finally, CVPDL seem to suffer from situations with changing backgrounds because of to its matching scheme, especially due to changes in shadows, while performing better in situations where the colour of the clothing is more uniform or distinct texture is visible in both camera views. Overall, information captured by each feature type positively affect late fusion as corresponding output lists differ. Few cases exist in which false identities are ranked high by all three systems as seen in Figure D.10, where lighting changes, uniform colours and similarities in the shape of the body, cause a false identify to be ranked higher.

4.4 Comparison to state-of-the-art

The proposed system is compared with other state-of-the-art systems on all four datasets. Tables D.3-D.6 contain the results for our system compared to the state-of-the-art. $Ours_{agg}$ and $Ours_{sco}$ indicate our system by fusion of both low-, mid-, and high-level features using rank aggregation and score-

level fusion, respectively.

Table D.3: *Comparison between our system and state-of-the-art systems on the VIPeR dataset (p=316). Best results are in bolt.* [1]

Method/Rank	r = 1	r = 5	r = 10
Ours_{agg}	45.63	75.06	85.16
Ours_{sco}	45.24	75.02	85.70
FFN+LOMO [28]	**51.06**	**81.01**	**91.39**
LOMO+XQDA [19]	40.00	67.40	80.51
Mirror-KMFA [37]	42.97	75.82	87.28
MuRE [30]	42.72	–	88.04
SCNCD [20]	37.80	68.50	81.20
Deep Re-id [25]	34.81	63.72	76.24
KISSME [13]	24.75	53.48	67.44
ECM [31]	38.90	67.80	78.40
MLF+LADF [10]	43.39	73.04	87.28

Table D.4: *Comparison between our system and state-of-the-art systems on the PRID450S dataset (p=225). Best results are in bolt.* [1]

Method/Rank	r = 1	r = 5	r = 10
Ours_{agg}	**77.56**	**93.47**	**96.09**
Ours_{sco}	68.76	88.49	93.47
FFN+LOMO [28]	66.62	86.84	92.84
Mirror-KMFA [37]	55.42	63.72	87.72
KISSME [13]	36.31	65.11	75.42
ECM [31]	41.90	66.30	76.90
SCNCD [20]	41.60	68.90	79.40

It is clearly shown that our introduced fusion results in a system that outperforms previous systems. For VIPeR, the feature fused system of LOMO and FFN achieves a rank-1 accuracy 5.43% better than the proposed system indicating that FFN and CVPDL might share the same difficulties when classifying certain identities which is also indicated by their performance being very similar. Compared to the related system of [30], our system achieves a rank-1 accuracy which is 2.91% higher, showing the importance of training on features at different abstraction levels.

For PRID450S and CUHK01 our system clearly beats the feature fused systems by a rank-1 increase of 10.94% and 14.84%, respectively.

For CUHK03, our results outperforms the rank-1 accuracy of [19] with 2.52% while having almost similar accuracy compared to the state-of-the-art CNN of [25]. Looking at Figure D.7 (d), this is probably due to the perfor-

Table D.5: *Comparison between our system and state-of-the-art systems on the CUHK01 dataset (p=486). Best results are in bolt.* [1]

Method/Rank	r = 1	r = 5	r = 10
Ours_{agg}	**70.35**	**88.46**	**93.29**
Ours_{sco}	64.67	83.81	89.35
FFN+LOMO [28]	55.51	78.40	83.68
Deep Re-id [25]	47.53	72.10	80.53
Mirror-KMFA [37]	40.40	64.63	75.34
MLF [10]	34.30	55.12	64.91

Table D.6: *Comparison between our system and state-of-the-art systems on the CUHK03 dataset (p=100). Best results are in bolt.* [1]

Method/Rank	r = 1	r = 5	r = 10
Ours_{agg}	52.25	80.40	88.95
Ours_{sco}	54.72	81.25	89.40
LOMO+XQDA [19]	52.20	82.23	**92.14**
Deep Re-id [25]	**54.74**	**86.42**	91.50
FPNN [5]	20.65	51.32	68.74

mance of FFN with a rank-1 accuracy of 20%. The CNN by Ahmed *et al.* was trained on CUHK03 while FFN was trained on the Market-1501 dataset [6] which, along with the architecture, might be the reason for the almost similar performance.

4.5 Cross-dataset test

In a real world scenario, it is desired to have a system that adapts well to new data. Furthermore, it is undesired to label new training data in each new application. As a result, a decent accuracy is desired, independently of which dataset is used for training. To test this scenario, training and testing are performed on different datasets.

For training, the extended CUHK01, CUHK02 [42], is utilized. This dataset consists of 1816 different identities, each with two images in two different views, bringing the total number of images to 7264. In the training phase, all identities are included using one image from each view. In the test, VIPeR is used, using the same identities in each iteration as in the intra-dataset test. Similar to the previous tests, 10 iterations are run and the accuracies are averaged. The resulting CMC curves are shown in Figure D.11. As in the test on CUHK03, score-level fusion provides the highest accuracies, having a rank-1 accuracy 5.95% higher than rank aggregation. Compared to the individual results, dominated by LOMO, the rank-1 accuracy is increase by 10.95%, once

again showing the benefit of late fusion.

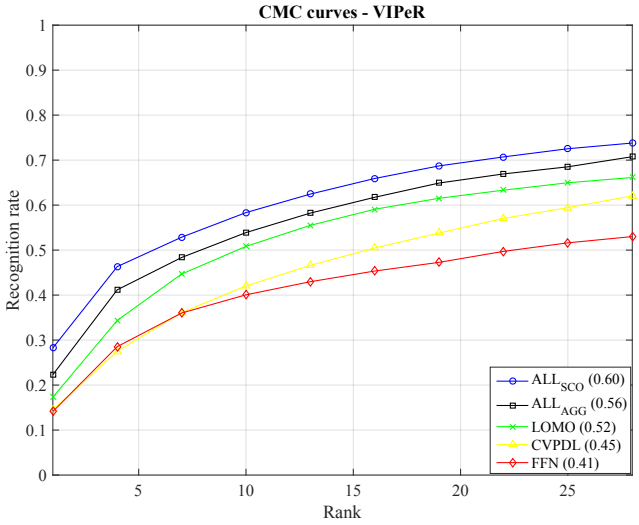


Fig. D.11: CMC curves of results using cross-dataset settings on VIPeR (p=316).

Table D.7 shows our results compared to previous systems tested under similar settings. Both previous systems utilize CNN's to learn similarity between identities. Our fusion system increase the rank-1 accuracy by 5.85% compared to the previous best result, despite the fact that the two other systems utilize multiple images from each identity in the training phase compared to only one in our case.

Table D.7: *Comparing to other state-of-the-art results when training on CUHK02 and testing on VIPeR (p=316). "–" indicates non available results. Best results are in bold.* [1]

System/Rank	r = 1	r = 5	r = 10	r = 20
Ours$_{agg}$	22.31	43.89	53.89	65.66
Ours$_{sco}$	**28.26**	**49.10**	**58.34**	69.35
DeepRank [43]	22.41	–	56.39	**72.72**
DML [26]	16.17	–	45.82	57.56

4.6 Processing time

In order to analyse whether it makes sense to make use of the introduced late fusion, the processing time when matching for each individual system along with the processing time for the fusion techniques are examined and compared to the increase in the accuracy.

A test is made by averaging the timings of 10 iterations on the VIPeR dataset using an Intel i7-4700MQ CPU @ 2.4GHz and the results are shown in Table D.8.

Table D.8: Average timings for matching and late fusion in seconds over 10 iterations on VIPeR dataset.

FFN	LOMO	CVPDL	Aggregation	Score-level
10.30	0.18	292.38	6.26	0.12

As shown in Table D.8, matching for CVPDL takes up 96.5% of the total processing time if score-level fusion is employed. In reality, this might not be suitable if the system is running real-time and a different way of matching patches should be developed if the algorithm should be kept in the fusing system.

Furthermore, rank aggregation and score-level fusion only takes up 2.1% and 0.04%, respectively, of the total processing time.

5 Conclusion

Throughout this paper, we have proposed a novel method to combine three state-of-the-art features developed for re-identification through late fusion. In order to get the most proper results, the features are extracted at different abstraction levels, namely low-, mid- and high-level. Two types of late fusion techniques are utilized, score-level fusion and rank aggregation, focusing on different ways to fuse outputs. The score-level fusion is re-defined to fit the scores we calculate.

Experimental results on four diff erent datasets showed a clear improvement in rank-1 accuracies on two with rank-1 accuracies of 10.94% and 14.84% for PRID450S and CUHK01, respectively, compared to previous systems and an increase on VIPeR of 5.43% compared to related systems. The results on CHUK03 are almost similar in performance compared to the state-of-the-art CNN of [25] with a potential to be increased by training or fine-tuning on similar dataset. Overall, rank aggregation provided the best results being up to 8.8% better than score-level fusion, though, with the latter being faster in processing time. Further, an analysis indicated that rank aggregation performed better when individual results are within a certain range of one another while score-level fusion is better when one result is much worse than the other two. In addition, the analysis showed that FFN mostly captures the overall contour of the body, LOMO mostly the combination of colours while CVPDL mostly captures the texture. The different focus points cause different ordering of matches which positively affect late fusion when the true

match is ranked high. When looking at the processing time, late fusion only takes up at most 2.1% of the total processing time, making late fusion beneficial when compared to the increased accuracies. Finally, patch matching showed a large increase in processing time, leaving it to be modified if kept in the fusing system.

References

[1] A. R. Lejbølle, "Enhancing person re-identification by late fusion," Master's thesis, Aalborg University, Denmark, 2016.

[2] Z. Liu, Z. Zhang, Q. Wu, and Y. Wang, "Enhancing person re-identification by integrating gait biometric," *Neurocomputing*, vol. 168, pp. 1144 – 1156, 2015.

[3] B. DeCann and A. Ross, "Modelling errors in a biometric re-identification system," *IET Biometrics*, vol. 4, no. 4, pp. 209–219, 2015.

[4] A. Schumann and E. Monari, "A soft-biometrics dataset for person tracking and re-identification," in *Proc. AVSS*, 2014, pp. 193–198.

[5] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[6] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. ICCV*, 2015, pp. 1116–1124.

[7] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[8] L. Bazzani, M. Cristani, and V. Murino, "Symmetry-driven accumulation of local features for human characterization and re-identification," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 130–144, 2013.

[9] S. Bak and F. Brémond, "Re-identification by covariance descriptors," in *Person Re-Identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 4, pp. 71–91.

[10] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-identification," in *Proc. CVPR*, 2014, pp. 144–151.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, vol. 2, 2006, pp. 2169–2178.

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

[13] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[14] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. ECCV*, 2014, pp. 1–16.

[15] M. Pantic and L. J. Rothkrantz, "Toward an affect-sensitive multimodal human-computer interaction," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1370–1390, 2003.

[16] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[17] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proc. WWW*, 2001, pp. 613–622.

[18] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Proc. CVPR*, 2013, pp. 3586–3593.

[19] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[20] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *Proc. ECCV*, 2014, pp. 536–551.

[21] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary, "Person re-identification by support vector ranking," in *Proc. BMVC*, 2010, pp. 21.1–21.11.

[22] W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *Proc. CVPR*, 2011, pp. 649–656.

[23] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, and Q. Tian, "Person re-identification meets image search," *CoRR*, vol. abs/1502.02171, pp. 4321–4330, 2015, retrieved from: http://arxiv.org/abs/1502.02171. [Online]. Available: http://arxiv.org/abs/1502.02171

[24] X. Liu, M. Song, D. Tao, X. Zhou, C. Chen, and J. Bu, "Semi-supervised coupled dictionary learning for person re-identification," in *Proc. CVPR*, 2014, pp. 3550–3557.

[25] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. CVPR*, 2015, pp. 3908–3916.

[26] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, 2014, pp. 34–39.

[27] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Proc. CVPR*, 2015, pp. 1846–1855.

[28] S. Wu, Y.-C. Chen, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. WACV*, 2016, pp. 1–8.

[29] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. CVPR*. IEEE, 2015, pp. 1741–1750.

[30] N. Martinel, C. Micheloni, and G. L. Foresti, "A pool of multiple person re-identification experts," *Pattern Recognition Letters*, vol. 71, pp. 23–30, 2016.

[31] X. Liu, H. Wang, Y. Wu, J. Yang, and M.-H. Yang, "An ensemble color model for human re-identification," in *Proc. WACV*, 2015, pp. 868–875.

[32] M. Ye, J. Chen, Q. Leng, C. Liang, Z. Wang, and K. Sun, "Coupled-view based ranking optimization for person re-identification," in *Proc. MMM*, 2015, pp. 105–117.

[33] R. F. de Carvalho Prates and W. R. Schwartz, "Cbra: Color-based ranking aggregation for person re-identification," in *Proc. ICIP*. IEEE, 2015, pp. 1975–1979.

[34] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[35] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. CVPR*, 2010, pp. 1301–1306.

[36] S. Li, M. Shao, and Y. Fu, "Cross-view projective dictionary learning for person re-identification," in *Proc. AAAI*. AAAI Press, 2015, pp. 2155–2161.

[37] Y.-C. Chen, W.-S. Zheng, and J. Lai, "Mirror representation for modeling view-specific transform in person re-identification," in *Proc. IJCAI*. Citeseer, 2015, pp. 3402–3408.

[38] R. F. de Carvalho Prates and W. R. Schwartz, "Appearance-based person re-identification by intra-camera discriminative models and rank aggregation," in *Proc. ICB*, 2015, pp. 65–72.

[39] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.

[40] R. Kolde, S. Laur, P. Adler, and J. Vilo, "Robust rank aggregation for gene list integration and meta-analysis," *Bioinformatics*, vol. 28, no. 4, pp. 573–580, 2012.

[41] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Person Re-Identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 12, pp. 247–267.

[42] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning." in *Proc. ACCV*. Springer, 2012, pp. 31–44.

[43] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2353–2367, 2016.

Part IV

Practical Re-id

Paper E

Camera On-boarding for Person Re-identification using Hypothesis Transfer Learning

Sk Miraj Ahmed, Aske R. Lejbølle, Rameswar Panda, and Amit K. Roy-Chowdhury

Submitted to the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020

The layout has been revised.

Abstract

Most of the existing approaches for person re-identification consider a static setting where the number of cameras in the network is fixed. An interesting direction, which has received little attention, is to explore the dynamic nature of a camera network, where one tries to adapt the existing re-identification models after on-boarding new cameras, with little additional effort. There have been a few recent methods proposed in person re-identification that attempt to address this problem by assuming the labeled data in the existing network is still available while adding new cameras. This is a strong assumption since there may exist some privacy issues for which one may not have access to those data. Rather, based on the fact that it is easy to store the learned re-identifications models, which mitigates any data privacy concern, we develop an efficient model adaptation approach using hypothesis transfer learning that aims to transfer the knowledge using only source models and limited labeled data, but without using any source camera data from the existing network. Our approach minimizes the effect of negative transfer by finding an optimal weighted combination of multiple source models for transferring the knowledge. Extensive experiments on four challenging benchmark datasets with variable number of cameras well demonstrate the efficacy of our proposed approach over state-of-the-art methods.

1 Introduction

Person re-identification (re-id), which addresses the problem of matching people across different cameras, has attracted intense attention in recent years [1, 2]. Much progress has been made in developing a variety of methods to learn features [3–5] or distance metrics by exploiting unlabeled and/or manually labeled data. Recently, deep learning methods have also shown significant performance improvement on person re-id [6–11]. However, with the notable exception of [12, 13], most of these works have not yet considered the dynamic nature of a camera network, where new cameras can be introduced at any time to cover a certain related area that is not well-covered by the existing network of cameras. To build a more scalable person re-identification system, it is very much essential to consider the problem of how to on-board new cameras into an existing network with little additional effort.

Let us consider K number of cameras in a network for which we have learned $\binom{K}{2}$ number of optimal pairwise matching metrics, one for each camera pair (see Fig. E.1 for an illustrative example). However, during an operational phase of the system, new camera(s) may be temporarily introduced to collect additional information, which ideally should be integrated with minimal effort. Given newly introduced camera(s), most of the prior re-id methods aim to re-learn the pairwise matching metrics using a costly training phase. This is impractical in many situations where the newly added

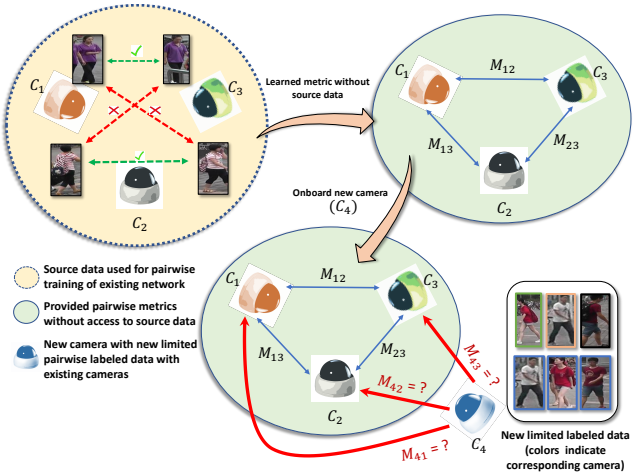


Fig. E.1: Consider a three camera (C_1, C_2 and C_3) network, where we have only three pairwise distance metrics (M_{12}, M_{23} and M_{13}) available for matching persons, and no access to the labeled data due to privacy concerns. During an operational phase of the system, a new camera, C_4, is installed into the system where we have very limited labeled data across the new camera and the existing ones. Our goal in this paper is to learn the pairwise distance metrics between the newly inserted camera(s) and the existing cameras (M_{41}, M_{42} and M_{43}), using the learned source metric from the existing network, with a small amount of labeled data available after installing the new camera(s).

camera(s) need to be operational soon after they are added. In this case, we cannot afford to wait a long time to obtain significant amount of labeled data for learning pairwise metrics, thus, we only have limited labeled data of persons that appear in the entire camera network after addition of the new camera(s).

Recently published works [12, 13] attempt to address the problem of integrating new cameras to a network by utilizing old data that were collected in the original camera network, combined with newly collected data in the expanded network and source metrics to learn new pairwise metrics. They also assume that the same set of people appears in all camera views, including the new camera (i.e., before and after on-boarding new cameras) for measuring the view similarity. However, this is unrealistic in many surveillance scenarios as source camera data may have been lost or not accessible because of some privacy concerns. Additionally, new people may appear after the target camera is installed who may or may not have appeared in existing source cameras. Motivated by this observation, we pose an important question: *How can we swiftly on-board new camera(s) in an existing re-id framework (i) without having access to the source camera data that the original network was trained on, and (ii) relying upon only a small amount of labeled data after adding the new camera(s).*

Transfer learning, which focuses on transferring knowledge from a source to a target domain, has recently been very successful in various computer vision problems [14–18]. However, knowledge transfer in our system is challenging, because of limited labeled data and absence of source camera data while on-boarding new cameras. To solve these problems, we develop an efficient model adaptation approach using *hypothesis transfer learning* that aims to transfer the knowledge using only source models (i.e., learned metrics) and limited labeled data, but without using any source camera data. Only a few labeled identities that are seen by the target camera, and one or more of the source cameras, are needed for effective transfer of source knowledge to the newly introduced target cameras. Henceforth, we will refer to this as *target data*. Furthermore, unlike [12, 13], which find only one best source camera that aligns maximally with the target camera, our approach focuses on finding an optimal weighted combination of multiple source models for transferring the knowledge.

Our approach works as follows. Given a set of pairwise source metrics and limited labeled target data after adding the new camera(s), we develop an efficient convex optimization formulation based on hypothesis transfer learning [19, 20] that tries to minimize the effect of negative transfer from any outlier source metric while transferring knowledge from source to the target cameras. More specifically, we learn the weights of different source metrics and the optimal matching metric jointly by alternating minimization, where the weighted source metric is used as a biased regularizer that aids to learn the optimal target metric only using limited labeled data. The proposed method, essentially, learns which camera pairs in the existing source network best describe the environment that is covered by the new camera and one of the existing cameras. Note that, our approach can be easily extended to multiple additional cameras being introduced at a time in the network or added sequentially one after another.

1.1 Contributions

We address the problem of swiftly on-boarding new camera(s) into an existing person re-identification framework without having access to the source camera data, and relying upon only a small amount of labeled data after adding the new cameras. Towards solving the problem, our contributions in this paper are the following.

- We propose a robust and efficient multiple metric hypothesis transfer learning algorithm to efficiently adapt a newly introduced camera to an existing person re-id framework without having access to the source data.
- We theoretically analyse the properties of our algorithm and show that

it minimizes the risk of negative transfer and performs closely to fully supervised case even when a small amount of labeled data is available.

- We perform rigorous experiments on multiple benchmark datasets to show the effectiveness of our proposed approach over existing alternatives.

2 Related Works

Person Re-identification. Most of the methods in person re-id are based on supervised learning. These methods apply extensive training using lots of manually labeled training data, and can be broadly classified in two categories: (i) *Distance metric learning based* [3, 21–25] (ii) *Deep learning based* [6, 10, 11, 26–29]. *Distance metric learning based* methods tend to learn distance metrics for camera pairs using pairwise labeled data between those cameras, whereas end to end *Deep learning based* methods tend to learn robust feature representations of the persons, taking into consideration all the labeled data across all the cameras at once. To overcome the problem of manual labeling, several unsupervised [14, 25, 30–33] and semi-supervised [34–37] methods have been developed over the past decade. However, these methods do not consider the case where new cameras are added to an existing network. The most recent approach in this direction [12, 13] has considered unsupervised, as well as semi-supervised, domain adaptation of the target camera by making a strong assumption of accessibility of the source data. None of the methods have used the fact of not having access to the source data in the dynamic camera network setting. This is relevant, as source data might have to be deleted after a while due to privacy issues.

Hypothesis Transfer Learning. Hypothesis transfer learning [19, 20, 38–40] is a type of transfer learning that uses only the learned classifiers from a source domain to efficiently learn a classifier in the target domain, which contains only limited labeled data. This approach is practically appealing as it does not assume any relatedness between source and target distribution nor any the availability of source data, which may be non accessible due to security reason [19]. Most of the literature has dealt with simple linear classifiers for transferring knowledge [19, 41]. One recent works [42] has addressed the problem of transferring the knowledge of a source metric, which is a positive semi-definite matrix, with some provable guarantees. However, it has been analyzed for only a single source metric and the weight of the metric is calculated by minimizing a cost function using sub-gradient descent from the generalization bound separately, which is a highly non-convex non-differential function. In [41], the method has addressed transfer of multiple linear classifiers in an SVM framework, where the corresponding weights are

calculated jointly with the target classifiers in a single optimization. Unlike these approaches, our approach addresses the case of transfer from multiple source metric hypotheses by jointly optimizing for target metric, as well as the corresponding weights to reduce the risk of negative transfer.

3 Methodology

Let us consider a camera network with K number of cameras for which we have learned a total $N = \binom{K}{2}$ pairwise metrics using extensive labeled data. We wish to install some new camera(s) in the system that need to be operational soon after they are added, i.e., without collecting and labeling lots of new training data. Due to privacy concerns, we do not have access to the old source camera data, rather, we only have the pairwise source distance metrics. Moreover, we also have access to only a limited amount of labeled data across the target and different source cameras, which is collected after installing the new cameras. Using the source metrics and the limited pairwise source-target labeled data, we propose to solve a constrained convex optimization problem (Eq. E.1) that aims to transfer knowledge from the source metrics to the target efficiently while minimizing the risk of negative transfer.

Formulation. Suppose we have access to the optimal distance metric $M_{ab} \in \mathbb{R}^{d \times d}$ for the a and b-th camera pair of an existing re-id network, where d is the dimension of the feature representation of the person images and $a, b \in \{1, 2 \dots K\}$. We also have limited pairwise labeled data $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$ between the target camera τ and the source camera p, where $x_{ij} = (x_i - x_j)$ is the feature difference between person i in camera τ and person j in camera p. $C = \binom{n_{\tau p}}{2}$, where $n_{\tau p}$ is the total number of different persons who appeared on the cameras τ and p, and $y_{ij} \in \{-1, 1\}$. $y_{ij} = 1$ if the persons i and j are the same person across the cameras, and -1 otherwise. Note that our approach does not need the presence of every person seen in the new target camera across all the source cameras, rather, in at least one of the source camera while computing the new distance metric across source-target pairs. Let S and D be defined as $S = \{(i, j) \mid y_{ij} = 1\}$ and $D = \{(i, j) \mid y_{ij} = -1\}$. Our main goal is to learn the optimal metric between target and each of the source cameras by using the information from all the pairwise source metrics $\{M_j\}_{j=1}^{N}$ and limited labeled data $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$. In standard metric learning context, the distance between two feature vectors $x_i \in \mathbb{R}^d$ and $x_j \in \mathbb{R}^d$ with respect to a metric $M \in \mathbb{R}^{d \times d}$ is calculated by $\sqrt{(x_i - x_j)^\top M (x_i - x_j)}$.

Thus, we formulate the following optimization problem for calculating the optimal metric $M_{\tau p}$ between target camera τ and the p-th source camera,

with n_s and n_d number of similar and dissimilar pairs, as follows:

$$\begin{aligned} \underset{M_{\tau p}}{\text{minimize}} \quad & \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M_{\tau p} x_{ij} + \lambda \|M_{\tau p} - \sum_{j=1}^{N} \beta_j M_j\|_F^2 \\ \text{subject to} \quad & \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M_{\tau p} x_{ij}) - b \geq 0,\ M_{\tau p} \succeq 0, \\ & \beta \geq 0,\ \|\beta\|_2 \leq 1 \end{aligned} \tag{E.1}$$

The above objective consists of two main terms. The first term is the normalized sum of distances of all similar pair of features between camera τ and p with respect to the Mahalanobis metric $M_{\tau p}$, and the second term represents the frobenius norm of the difference of $M_{\tau p}$ and weighted combination of source metrics squared. λ is a regularization parameter to balance the two terms. Note that the second term in Eq. E.1 is essentially related to hypothesis transfer learning [19, 20] where the hypotheses are the source metrics. The first constraint represents that the normalized sum of distances of all dissimilar pairs of features with respect to $M_{\tau p}$ is greater than a user defined threshold b, and the second constraints the distance metrics to always lie in the positive semi-definite cone. While the third constraint keeps all the elements of the source weight vector non-negative, the last constraint ensures that the weights should not deviate much from zero (through upper-bounding the ℓ-2 norm by 1).

Notation. We use the following notations in the optimization steps.

(a) $\mathcal{C}_1 = \{M \in \mathbb{R}^{d\times d} \mid \sum_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0\}$

(b) $\mathcal{C}_2 = \{M \in \mathbb{R}^{d\times d} \mid M \succeq 0\}$

(c) $\mathcal{C}_3 = \{\beta \in \mathbb{R}^N \mid \beta \geq 0 \cap \|\beta\|_2 \leq 1\}$

Optimization. The proposed optimization problem (E.1) is convex. To handle the optimization over large size matrices without memory issues, we devise an iterative algorithm to efficiently solve (E.1) by alternately solving for two sub-problems. For the sake of brevity, we denote $M_{\tau p}$ as M in the subsequent steps. Specifically, in the first step, we fix the weight β and take a gradient step with respect to M in the descent direction with step size α (Eq. E.2). Then, we project the updated M onto $\mathcal{C}_1$ and $\mathcal{C}_2$ in an alternating fashion until it converges (Eq. E.3 and Eq. E.4). In the next step, we fix the the updated M and take a step with size γ towards the direction of negative gradient with respect to β (Eq. E.6). In the last step, we simply project β onto the set $\mathcal{C}_3$ (Eq. E.7). Algorithm 1 summarizes the alternating minimization procedure to optimize (E.1). We briefly describe these steps below and refer the reader to the supplementary material for more mathematical details.

Algorithm 1: Algorithm to Solve Eq. E.1

Input: Source metric $\{M_j\}_{j=1}^{N}$, $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$
Output: Optimal metric $M^\star$
Initialization: $M^k, \beta^k, k = 0$;
while *convergence* **do**
  $M^{k+1} = M^k - \alpha \nabla_M f(M, \beta^k)|_{M=M^k}$ (Eq. E.2);
  while *convergence* **do**
    $M^{k+1} = \Pi_{\mathcal{C}_1}(M^{k+1})$ (Eq. E.3);
    $M^{k+1} = \Pi_{\mathcal{C}_2}(M^{k+1})$ (Eq. E.4);
  end
  $\beta^{k+1} = \beta^k - \gamma \nabla_\beta (f(M^{k+1}, \beta)|_{\beta=\beta^k}$ (Eq. E.6);
  $\beta^{k+1} = \Pi_{\mathcal{C}_3}(\beta^k)$ (Eq. E.7);
  $k = k + 1$;
end

Step 1: Gradient w.r.t M with fixed β.
With k being the iteration number and M^k, β^k being M and β in the k-th iteration, we compute the gradient of the objective function (E.1) with respect to M by fixing $\beta = \beta^k$ at k-th iteration as follows:

$$\nabla_M f(M, \beta^k)|_{M=M^k} = \Sigma_S + 2\lambda(M^k - \sum_{j=1}^{N} \beta_j^k M_j), \tag{E.2}$$

where $\Sigma_S = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij} x_{ij}^\top$ and
$f(M, \beta^k) = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \lambda \|M - \sum_{j=1}^{N} \beta_j^k M_j\|_F^2$.

Step 2: Projection of M onto $\mathcal{C}_1$ and $\mathcal{C}_2$. The projection of M onto $\mathcal{C}_1$ (denoted as $\Pi_{\mathcal{C}_1}(M)$) can be computed by solving a constrained optimization as follows:

$$\begin{aligned} \Pi_{\mathcal{C}_1}(M) = &\arg\min_{\hat{M}} \quad \frac{1}{2}\|\hat{M} - M\|_F^2 \\ &\text{Subject to} \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top \hat{M} x_{ij}) - b \geq 0 \end{aligned}$$

By writing the Lagrange for the above constrained optimization and using KKT conditions with strong duality, the projection of M onto $\mathcal{C}_1$ can be

written as

$$\Pi_{\mathcal{C}_1}(M) = M + \max\left\{0, \frac{\left(b - \frac{1}{n_d}\sum\limits_{(i,j)\in D} x_{ij}^\top M x_{ij}\right)}{\|\Sigma_D\|_F^2}\right\}\Sigma_D, \tag{E.3}$$

where $\Sigma_D = \sum\limits_{(i,j)\in D} x_{ij}x_{ij}^\top$. Similarly, using spectral value decomposition, the projection of M onto $\mathcal{C}_2$ can be written as

$$\Pi_{\mathcal{C}_2}(M) = V diag(\begin{bmatrix}\hat{\lambda_1} & \hat{\lambda_2} \dots \hat{\lambda_n}\end{bmatrix})V^\top, \tag{E.4}$$

where V is the matrix constructed by stacking the eigenvectors of M column-wise, λ_i is the i-th eigenvalue of M and $\hat{\lambda_j} = \max\{\lambda_j, 0\} \quad \forall \quad j \in \begin{bmatrix}1 \dots d\end{bmatrix}$.

Step 3: Gradient w.r.t β with fixed M. By fixing $M = M^{k+1}$ in the objective function, differentiating it w.r.t β_i, the i-th element of β at the point $\beta = \beta^k$, we get

$$\begin{aligned}\nabla_{\beta_i}(f(M^{k+1},\beta)|_{\beta_i=\beta_i^k} = 2\lambda\beta_i^k \text{trace}(M_i^\top M_i) - \\ 2\lambda \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j^k M_j))\end{aligned} \tag{E.5}$$

By denoting $\nabla_{\beta_i}(f(M^{k+1},\beta)|_{\beta_i=\beta_i^k}$ as a_i^k, we get

$$\nabla_\beta(f(M^{k+1},\beta)|_{\beta=\beta^k} = \begin{bmatrix}a_1^k & a_2^k & \dots & a_N^k\end{bmatrix}^\top \tag{E.6}$$

Step 4: Projection of β onto $\mathcal{C}_3$. This step essentially projects a vector to the first quadrant of an N-dimensional unit norm hyper-sphere. The closed form expression of the projection onto $\mathcal{C}_3$ is as follows:

$$\Pi_{\mathcal{C}_3}(\beta) = \max\left\{0, \frac{\beta}{\max\{1, \|\beta\|_2\}}\right\} \tag{E.7}$$

4 Discussion and Analysis

One of the key differences between our approach and existing methods is that the nature of our problem deals with the multiple metric setting within the hypothesis transfer learning framework. In this section, following [42], we theoretically analyze the properties of our algorithm 1 for transferring knowledge from multiple metrics.

Let $\mathcal{T}$ be a domain defined over the set $(\mathcal{X} \times \mathcal{Y})$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \in \{-1, 1\}$ denote the feature and label set, respectively, and has a probability distribution denoted by $\mathcal{D}_\mathcal{T}$. Let T be the target domain defined by

$\{(x_i, y_i)\}_{i=1}^{n}$ consisting of n i.i.d samples, each drawn from the distribution $\mathcal{D}_{\mathcal{T}}$. The optimization proposed in Eq.1 of [42] (page. 2) is defined as:

$$\underset{M \succeq 0}{\text{minimize}} \;\; L_T(M) + \lambda \|M - M_S\|_F^2 \tag{E.8}$$

Fixing the value of β in our proposed optimization (E.1), we have an optimization problem equivalent to (E.8), where $M_S = \sum_{j=1}^{N} \beta_j M_j$ and

$$L_T(M) = \frac{1}{n_s} \sum_{(i,j) \in S} x_{ij}^\top M x_{ij} + \mu^\star \big(b - \frac{1}{n_d} \sum_{(i,j) \in D} x_{ij}^\top M x_{ij}\big) \tag{E.9}$$

Note that $\mu^\star$ in Eq. E.9 is the optimal dual variable for the inequality constraint optimization (E.1) with the weight vector fixed. Clearly, the expression is linear, hence convex in M, and has a finite lipschitz constant k.

Theorem 1. *For the convex and k-Lipschitz loss (shown in supp) defined in* (E.9) *the average bound can be expressed as*

$$\mathbb{E}_{T \sim \mathcal{D}_{\mathcal{T}}{}^n} [L_{\mathcal{D}_{\mathcal{T}}}(M^\star)] \leq L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S}) + \frac{8k^2}{\lambda n}, \tag{E.10}$$

where n is the number of target labeled example, $M^\star$ is the optimal metric computed from Algorithm 1, $\widehat{M_S}$ is the average of all source metrics defined as $\frac{\sum_{j=1}^{N} M_j}{N}$, $\mathbb{E}_{T \sim \mathcal{D}_{\mathcal{T}}{}^n} [L_{\mathcal{D}_{\mathcal{T}}}(M^\star)]$ is the expected loss by $M^\star$ computed over distribution $\mathcal{D}_{\mathcal{T}}$ and $L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S})$ is the loss of average of source metrics computed over $\mathcal{D}_{\mathcal{T}}$.

Proof. The proof is given in supplementary material. □

Implication of Theorem 1: Since we transfer knowledge from multiple source metrics, and do not know which is the most generalizable over the target distribution (i.e the best source metric), the most sensible thing is to check for the average performance of using each of the source metrics directly over the target test data. It is equivalently giving all the source metrics equal weights and not using any of the target data for training purpose. The bound in Theorem (E.9) shows that, on average, the metric learned form Algorithm 1 tends to do better than, or in worst case, at least equivalent to the average of source metrics with a fast convergence rate of $\mathcal{O}(\frac{1}{n})$ with limited number of target samples [42].

Negative Transfer: In optimization E.1, we jointly estimate the optimal metric, as well as the weight vector, which determines which source to transfer from and with how much weight. If a source metric does not generalize well across the target distribution, giving more weight to that metric will increase the loss in Eq. E.9. It is evident that our proposed optimization automatically prevents this increment since we have a constraint, where the weights can be

zero. Thus, our approach minimizes the risk of negative transfer by assigning the outlier source metrics weights that are as low as possible to reduce the loss defined in Eq. E.9.

Theorem 2. *With probability* $(1-\delta)$*, for any metric* M *learned from Algorithm 1 we have,*

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}(\frac{1}{n}) + \left(\sqrt{\frac{L_T(\sum_{j=1}^{N}\beta_j M_j)}{\lambda}} + \| \sum_{j=1}^{N} \beta_j M_j \|_F\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{E.11}$$

where $L_{\mathcal{D}_\mathcal{T}}(M)$ *is the loss over the original target distribution (true risk),* $L_T(M)$ *is the loss over the existing target data (empirical risk), and* n *is the number of target samples.*

Proof. See the supplementary material for the proof. □

Implication of Theorem 2: This bound shows that given only a small amount of labeled target data, our method performs closely to the fully supervised case, if some of the source metrics generalize well for the target distribution. The right hand side of the inequality (E.11) consists of the sum of the terms $\mathcal{O}(\frac{1}{n})$ and $\mathcal{O}(\frac{1}{\sqrt{n}})$ multiplied by a coefficient. Since the optimal weight $\beta^\star$ from optimization (E.1) will be sparse due to the way β is constrained and the zero coefficients will automatically be assigned to the outlier metrics resulting in smaller value of the coefficient associated with $\mathcal{O}(\frac{1}{\sqrt{n}})$. As a result the $\mathcal{O}(\frac{1}{n})$ term will be more dominant in (E.11) over $\mathcal{O}(\frac{1}{\sqrt{n}})$. Due to the faster decay rate of $\mathcal{O}(\frac{1}{n})$, this implies that with very limited target data, the empirical risk will converge to the true risk. Furthermore, when n is very large (the fully supervised case), $\mathcal{O}(\frac{1}{\sqrt{n}})$ will be close to zero and cannot be altered by multiplication with any coefficient. This implies that the source metrics will not have any effect on learning when there is enough labeled target data available and are only useful in the presence of limited data as in our application domain.

5 Experiments

Datasets. We test the effectiveness of our method by experimenting on four publicly available re-id datasets such as WARD [43], RAiD [44], Market1501 [45], and MSMT17 [46]. There are several other re-id datasets like ViPeR [47], PRID2011 [48] and CUHK01 [49], however, those do not apply in our case due to availability of only two cameras. RAiD and WARD are smaller datasets with 43 and 70 persons captured in 4 and 3 cameras, respectively, whereas

Market1501 and MSMT17 are more recent and large datasets with 1,501 and 4,101 persons captured across 6 and 15 cameras, respectively.

Feature Extraction and Matching. We use Local Maximal Occurrence (LOMO) feature [3] of length 29,960 in RAiD and WARD datasets. However, since LOMO usually performs poorly in large datasets [2], for Market1501 and MSMT17 we extract features from the last layer of a Imagenet [50] pretrained ResNet50 network [51] (denoted as IDE features in our work). We follow standard PCA technique to reduce the feature dimension to 100, as in [12, 23].

Performance Measures. We provide standard Cumulative Matching Curves (CMC) and normalized Area Under Curve (nAUC), as is common in person re-id [3, 13, 23, 44]. While the former shows accumulated accuracy by considering the k-most similar matches within a ranked list, the latter is a measure of re-id accuracy, independent on the number of test samples. Due to the space constraint, we only report average CMC curves for most experiments and leave the full CMC curves in the supplementary material.

Experimental Settings. For RAiD we follow the protocol in [3] and randomly split the persons into a training set of 21 persons and a test set of 20 persons, whereas for WARD, we randomly split the 70 persons into a set of 35 persons for training and rest 35 persons for testing. For both datasets, we perform 10 train/test splits and average accuracy across all splits. We use the standard training and testing splits for both Market1501 and MSMT17 datasets. During testing, We follow a multi-query approach by averaging all query features of each id in the target camera and compare with all features in the source camera [45].

Compared Methods. We compare our approach with the following methods. (1) Two variants of Geodesic Flow Kernel (GFK) [52] such as Direct-GFK where the kernel between a source-target camera pair is directly used to evaluate the accuracy and Best-GFK where GFK between the best source camera and the target camera is used to evaluate accuracy between all source-target camera pairs as in [12, 13]. Both methods use the the supervised dimensionality reduction method, Partial Least Squares (PLS), to project features into a low dimensional subspace [12, 13]. (2) state-of-the-art method for onboarding new cameras [12, 13] that uses transitive inference over the learned GFK across the best source and target camera. We refer this method as Adapt-GFK in our work. (3) Clustering-based Asymmetric MEtric Learning (CAMEL) method of [25], which projects features from source and target camera to a shared space using a learned projection matrix. For all compared methods, we use their publicly available code and perform evaluation in our setting.

5.1 On-boarding a Single New Camera

We consider one camera as newly introduced target camera and all the other as source cameras. We consider all the possible combinations for conducting experiments. In addition to the baselines described above, we compare against the accuracy of average of the source metrics (Avg-Source) by applying it directly over the target test set to prove the validity of Theorem 1. We also compute the GFK kernels in two settings; by considering only target data available after introducing the new cameras (Fig. E.2) and by considering the presence of both old source data and the new labeled data after camera installation as in [12, 13] (Fig. E.3).

Implementation details. We split training data into disjoint source and target data considering the fact that the persons that appear in the new camera after installation may or may not be seen before in the source cameras. That is, for Market1501 and MSMT17 we split the training data into 90% of persons that are only seen by the source cameras and 10% that are seen in both source cameras and the new target camera after the installation. Since there are much fewer persons in RAiD and WARD training set, we split the persons to 80% source and 20% target for those two datasets. For each dataset, we evaluate every source-target pair and average accuracy across all pairs. Furthermore, we average accuracy across all cameras as target. Note that the train and test set are kept disjoint in all our experiments.

Results. Fig. E.2 and E.3 show the results. In all cases, our method outperforms all the compared methods. The most competitive methods are those of Adapt-GFK and Avg-Source that also use source metrics. For the remaining methods, we see the limitation of only using limited target data to compute the new metrics. For Market1501, we see that Avg-Source outperforms the Adapt-GFK baseline indicating the advantage of knowledge transfer from multiple source metric compared to one single best source metric as in [12, 13]. However, our approach still outperforms the Avg-Source baseline by a margin of 20.60%, 13.81%, 2.01% and 1.07% in Rank-1 accuracy on RAiD, WARD, Market1501 and MSMT17, respectively, validating our implications of Theorem 1. Furthermore, we observe that even without accessing the source training data that was used for training the network before adding a new camera, our method outperforms the GFK based methods that use all the source data in their computations (see Fig. E.3). To summarize, the experimental results show that our method performs better on both small and large camera networks with limited supervision, as it is able to adapt multiple source metrics via minimization of negative transfer by dynamically weight the source metrics.

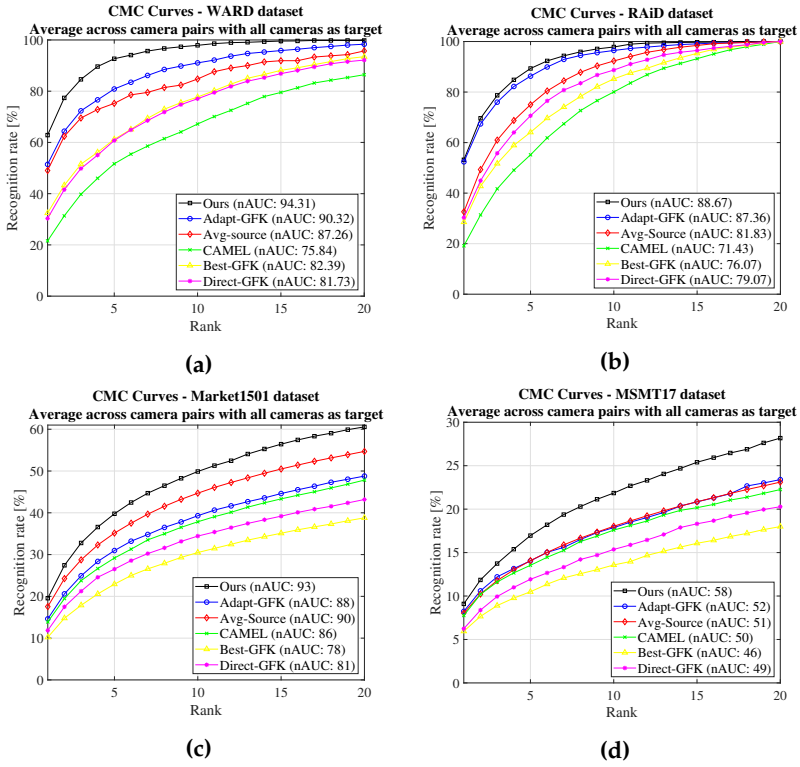


Fig. E.2: CMC curves averaged over all target camera combinations, introduced one at a time. (a) WARD with 3 cameras, (b) RAiD with 4 cameras, (c) Market1501 with 6 cameras and (d) MSMT17 with 15 cameras. Best viewed in color.

5.2 On-boarding Multiple New Cameras

We perform this experiment on Market1501 dataset using the same strategy as in Section 5.1 and compare our results with other methods while adding multiple target cameras to the network, either continuously or in parallel.

Parallel on-boarding of Cameras: We randomly select two or three cameras as target while keeping the remaining as source. All the new target cameras are tested against both source cameras and other target cameras. The results of adding two and three cameras in parallel (at the same time) are shown in Fig. E.4 (a) and (b), respectively. In both cases, our method outperforms all the compared methods with an increasing margin as rank increases. We outperform the most competitive baseline of CAMEL in Rank-1 accuracy by 5.45% and 3.73%, while adding two and three cameras respectively. Furthermore, our method better adapts source metrics since it has the capability of assigning zero weights to the metrics that do not generalize well over tar-

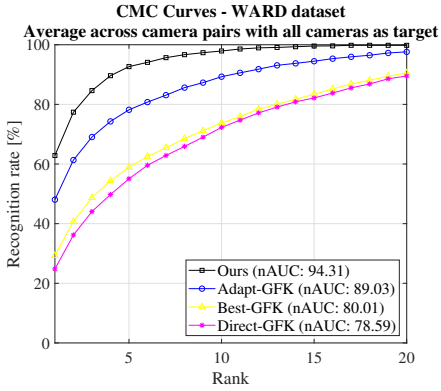


Fig. E.3: CMC curves averaged over all target camera combinations, introduced one at a time, on WARD dataset. Note that, both old and new source data are used for calculation of GFK. Best viewed in color.

get data. Meanwhile, Adapt-GFK has a high probability of using the outlier source metrics in the presence of fewer available source metrics, which causes negative transfer. This has been shown in Fig E.4 where GFK based methods are performing worse than CAMEL, which is computed just with limited supervision without using any source metrics.

Continual on-boarding of Cameras: For this experiment, we randomly select three target cameras that are added continuously. A target camera is tested against all source cameras and previously added target cameras. The results are shown in Fig. E.4 (c). Similar to parallel on-boarding, our methods outperforms compared methods by a large margin. In this setting, we outperform CAMEL by 8.22% in Rank-1 accuracy. Additionally, compared to all GFK-based methods, the Rank-1 margin is kept constant at 10% for both parallel and continuous on-boarding. These results show the scalability of our method while adding multiple cameras to a network, independent of whether they are added in parallel or continuously.

5.3 Different Labeled Data in New Cameras

We perform this experiment to show the implications of Theorem 2 by using different percentages of labeled target data (10%, 20%, 30%, 50%, 75% and 100%) in our method. We compare with a widely used KISS metric learning (KISSME) [23] algorithm and show the difference in Rank-1 accuracy as a function of labeled target data. Fig. E.5 (b) shows the results. At only 10% labeled data, the difference between our method and KISSME [23] is almost 30%, however, as we add more labeled data, the Rank-1 accuracy becomes equivalent for the two methods at 100% labeled data. This confirms the im-

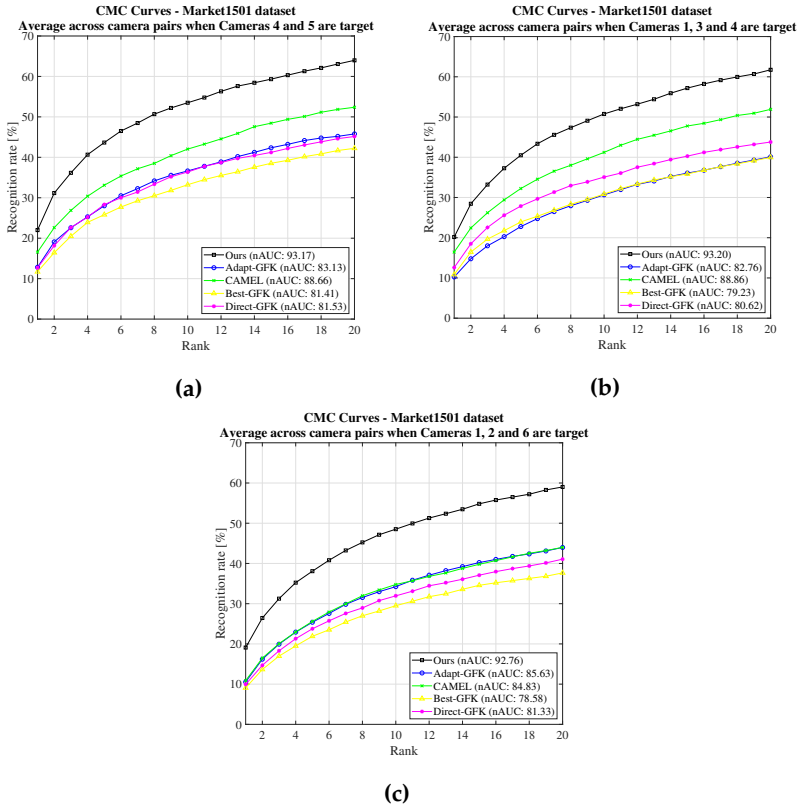


Fig. E.4: CMC curves averaged across target cameras on Market1501 dataset. (a) and (b) show results while adding two and three cameras simultaneously, (c) show the results while adding three cameras sequentially. Best viewed in color.

plications of Theorem 2, where we showed that with increasing labeled data in the target, the effect of source metrics in learning becomes negligible.

5.4 Finetuning with Deep Features

This experiment shows the effect of using features from a deep network that is first trained on the source data and later finetuned on new limited target data. We compare with supervised metric learning KISSME [23] and Eucldean distance metric using IDE features directly and using features after finetuning the model. We perform this experiment on Market1501 dataset using the same settings as described in Section 5.1. We train a ResNet50 model [51], pretrained on the Imagenet dataset, using the source data, and use the optimized source features to train the source metrics. Afterwards, we finetune the model using the labeled target data and use the new tar-

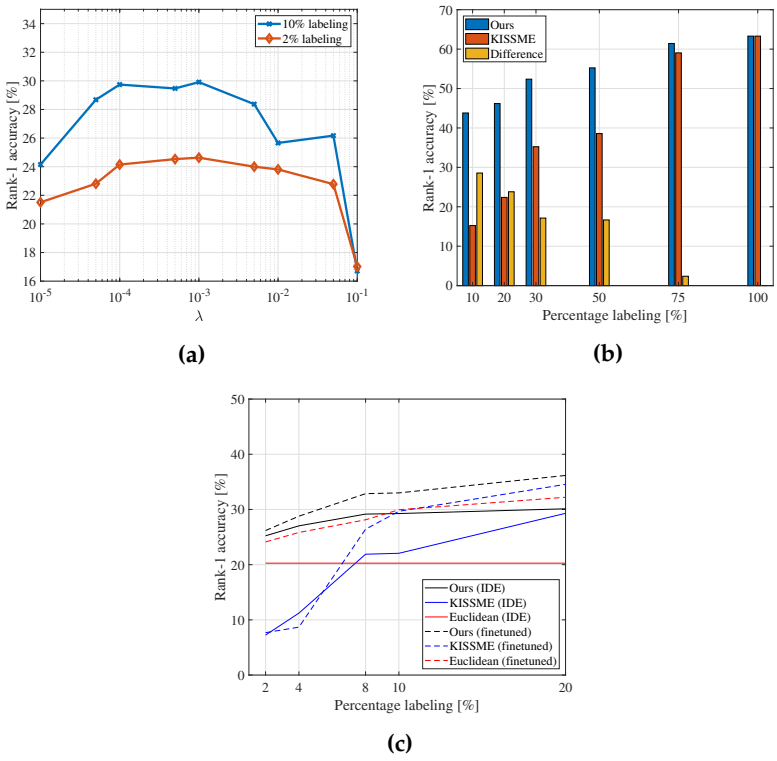


Fig. E.5: (a) Sensitivity of λ on the Rank-1 performance tested using deep features in Market1501 dataset with 6-th camera as target. (b) Effect of different percentage target labelling on ward dataset to compare with normal supervised method to justify theorem 2 (c) Analysis of our method with deep features trained on source camera data. best viewed in color.

get features along with the source metrics in optimization E.1. Please see supplementary material for more details. Fig. E.5 (c) shows the results with different percentage of labeled target data. At $< 8\%$ labeling, the differences between IDE and finetuned features are low for Ours and KISSME, however, we increase Rank-1 accuracy by more than 15% compared to KISSME in both cases. At 10% labeling, Ours (IDE) still performs better than both Euclidean (finetuned) and KISSME (finetuned). The accuracy of Ours (IDE) only increases slowly with more than 8% labeling, however, Ours (finetuned) still improves when increasing amount of labeled data. This indicates that our method works even better when using deep features from a model, which was finetuned on limited target data.

5.5 Parameter Sensitivity

We perform this experiment to study the effect of λ in optimization E.1 for a given percentage of labeled target data. Fig. E.5 (a) shows the Rank-1 accuracy of our proposed method accuracy for different values of λ. From optimization E.1, when $\lambda \to \infty$ the left term can be neglected resulting optimal M and β to be zero. However, when $\lambda \to 0$, the regularization term is neglected resulting in no transfer. We can see from Fig. E.5 that there is an operating zone of λ (e.g., in the range of 10^{-4} to 10^{-2}), that is neither too high nor too low for useful transfer from source metrics.

6 Conclusion

In this paper, we presented a simple yet effective model adaptation approach for person re-identification based on hypothesis transfer learning, which transfers knowledge using only learned source metrics and a limited amount of labeled data collected after installing the new cameras. We provided theoretical analysis to show that our approach minimizes the effect of negative transfer through finding an optimal weighted combination of multiple learned source metrics. We show the effectiveness of our proposed approach on four standard datasets, significantly outperforming several baseline methods.

Supplementary Material

Page number	Content
[188]	Dataset Descriptions
[189]	Detailed Description of the Optimization Steps
[193]	Proof of theorems from the main paper
[196]	On-boarding a Single new Camera (camera-wise cmc curves)
[201]	On-boarding multiple new Cameras (camera-wise cmc curves)
[202]	Finetuning with deep features

Table E.1: Supplementary Material Overview.

E.A Dataset Descriptions

This section contains detailed descriptions of the datasets used in our experiemnts (see Figure E.6 for sample images).

WARD [43] was collected from three outdoor cameras. The dataset contains 4,786 images of 70 different persons and includes variations in illumination.

RAiD [44] was collected from four cameras; two indoor and two outdoor. 6,920 images were captured of 43 different persons. However, two of these persons were only seen by two of the four cameras. As a result of having both indoor and outdoor cameras, the dataset includes large illumination and viewpoint variations.

Market1501 [45] was collected from six cameras and used a Deformable Part Model [53] to annotate images. This resulted in 32,668 images of 1,501 different persons, but also 2,793 "distractors" that are badly drawn bounding boxes. The dataset includes variations in both detection precision, resolution and viewpoint.

MSMT17 [46] is the largest person re-identification dataset to date, and contains images collected by no more than 15 cameras; 3 indoor and 12 outdoor. Data was collected over the course of four different days in a month, and Faster RCNN [54] was using for bounding box detection, resulting in 126,441 images of 4,101 different persons. Due to the diversity in data collection, this dataset contains large variations in illumination and viewpoint.

(a) WARD **(b)** RAiD **(c)** Market1501 **(d)** MSMT17

Fig. E.6: A total of 48 Sample images from the 4 datasets used in our experimentation. In each row 4 different persons are shown whereas for each column 3 different views of the same person from 3 different cameras are shown. We can see the that across cameras, the viewpoint of the same person is very diverse because of change in illumination condition or occlusion.

E.B Detailed Description of the Optimization Steps

In this section we will rigorously discuss all the necessary derivations of the steps of our proposed algorithm that could not be shown in the main paper due to space constraint. We first present the notations that we will use throughout this section.

Notations:

- $\frac{1}{n_s} \sum_{(i,j)\in S} x_{ij} x_{ij}^\top = \Sigma_S$
- $\frac{1}{n_d} \sum_{(i,j)\in D} x_{ij} x_{ij}^\top = \Sigma_D$
- $\mathcal{C}_1 = \{M \mid \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0\}$
- $\mathcal{C}_2 = \{M \mid M \succeq 0\}$
- $\mathcal{C}_3 = \{\beta \mid \|\beta\|_2 \leq 1\}$
- $\Pi_{\mathcal{C}}(X) = \underset{\hat{X}\in\mathcal{C}}{\text{minimize}} \ \frac{1}{2}\|\hat{X} - X\|_F^2$
- $f(M, \beta) = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \lambda \|M - \sum_{j=1}^N \beta_j M_j\|_F^2$

The proposed optimization problem in the main paper is defined below.

$$\begin{aligned} \underset{M_{\tau p}}{\text{minimize}} \quad & \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M_{\tau p} x_{ij} + \lambda \|M_{\tau p} - \sum_{j=1}^N \beta_j M_j\|_F^2 \\ \text{subject to} \quad & \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M_{\tau p} x_{ij}) - b \geq 0,\ M_{\tau p} \succeq 0, \\ & \beta \geq 0,\ \|\beta\|_2 \leq 1 \end{aligned} \tag{E.12}$$

Step 1: Gradient w.r.t M with fixed β.

$$\begin{aligned} \nabla_M (f(M, \beta)) &= \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij} x_{ij}^\top + 2\lambda (M - \sum_{j=1}^N \beta_j M_j) \\ &= \Sigma_S + 2\lambda (M - \sum_{j=1}^N \beta_j M_j) \end{aligned} \tag{E.13}$$

Step 2: Projection of M onto $\mathcal{C}_1$ and $\mathcal{C}_2$.

This can be done by solving a constrained optimization problem.

$$\Pi_{\mathcal{C}_1}(M) = \arg\min_{\hat{M}} \ \frac{1}{2}\|\hat{M} - M\|_F^2$$
$$\text{Subject to} \quad \frac{1}{n_d}\sum_{(i,j)\in D}(x_{ij}^\top \hat{M} x_{ij}) - b \geq 0$$

We can write the lagrangian as follows,

$$\mathcal{L}(\hat{M}, \psi) = \frac{1}{2}\|\hat{M} - M\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \hat{M} x_{ij}) \tag{E.14}$$

The KKT conditions for this problem are:

1.
$$\begin{aligned}
&\nabla_{\hat{M}}\mathcal{L}(\hat{M}, \psi)|_{\hat{M}=\hat{M}^\star} = 0 \\
&\implies (\hat{M}^\star - M) - \frac{\psi}{n_d}\sum_{(i,j)\in D} x_{ij}x_{ij}^\top = 0 \\
&\implies (\hat{M}^\star - M) - \psi\Sigma_D = 0 \\
&\implies \hat{M}^\star = (M + \psi\Sigma_D)
\end{aligned}$$

2. $\psi^\star(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \hat{M}^\star x_{ij}) \geq 0$

3. $\psi^\star \geq 0$

The optimization problem is convex, so strong duality should hold. So, we put the value of $\hat{M}^\star$ from KKT condition 1 in the equation (E.14) to get the

dual objective function as follows,

$$
\begin{aligned}
g(\psi) = \mathcal{L}(\hat{M}^\star, \psi) &= \frac{1}{2}\|M + \psi\Sigma_D - M\|_F^2 + \psi\big(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top (M + \psi\Sigma_D)x_{ij}\big) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \Sigma_D x_{ij} \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(x_{ij}^\top \Sigma_D x_{ij}) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(\Sigma_D x_{ij} x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij} x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D^\top \Sigma_D) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\|\Sigma_D\|_F^2 \\
&= -\frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})
\end{aligned}
\tag{E.15}
$$

To get the optimal $\psi^\star$ we have to maximize $g(\psi)$.

$$
\begin{aligned}
& g'(\psi^\star) = 0 \\
\implies & -\psi^\star\|\Sigma_D\|_F^2 + (b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) = 0 \\
\implies & \psi^\star = \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}
\end{aligned}
$$

But also from KKT condition (3), we know $\psi \geq 0$. Combining with the last equation we get

$$
\psi^\star = \max\left\{0, \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}\right\} \tag{E.16}
$$

So, putting the value of $\psi^\star$, finally we can write the projection from KKT

condition 1 as,

$$\Pi_{C_1}(M) = M + \max\left\{0, \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}\right\}\Sigma_D \tag{E.17}$$

projection onto $\mathcal{C}_2$ is standard, so we are not discussing it here.

Step 3: Gradient w.r.t β with fixed M.

$$\begin{aligned} f(M^{k+1},\beta) &= \frac{1}{n_s}\sum_{(i,j)\in S} x_{ij}^\top M^{k+1} x_{ij} + \lambda\|M^{k+1} - \sum_{j=1}^{N}\beta_j M_j\|_F^2 \\ &= K + \lambda\|M^{k+1} - \sum_{j=1}^{N}\beta_j M_j\|_F^2 \\ &= K + \lambda\text{trace}\Big((M^{k+1} - \sum_{j=1}^{N}\beta_j M_j)^\top (M^{k+1} - \sum_{j=1}^{N}\beta_j M_j)\Big) \\ &= K + \lambda\beta_i^2\text{trace}(M_i^\top M_i) - 2\lambda\beta_i\text{trace}(M_i^\top (M^{k+1} - \sum_{j=1,j\neq i}^{N}\beta_j M_j)) \end{aligned} \tag{E.18}$$

K is term which is independent of β. Now differentiating equation (E.18) w.r.t β_i we get ,

$$\nabla_{\beta_i} f(M^{k+1},\beta) = 2\lambda\beta_i\text{trace}(M_i^\top M_i) - 2\lambda\text{trace}(M_i^\top (M^{k+1} - \sum_{j=1,j\neq i}^{N}\beta_j M_j)) = a_i \tag{E.19}$$

So, derivative of $f(M^{k+1},\beta)$ w.r.t β is given by,

$$\nabla_\beta f(M^{k+1},\beta) = \begin{bmatrix} a_1 & a_2 & \dots & a_N \end{bmatrix}^\top \tag{E.20}$$

Step 4: Projection of β onto $\mathcal{C}_3$.

$$\Pi_{\mathcal{C}_3}(\beta) = \max\left\{0, \frac{\beta}{\max\{1,\|\beta\|_2\}}\right\} \tag{E.21}$$

The intuition here is that, when the norm of β is greater than 1 then $\max\{1,\|\beta\|_2\} = \|\beta\|_2$ which implies the normalization of β. Similarly when the norm of β is lesser or equal to 1 then $\max\{1,\|\beta\|_2\} = 1$, which means keeping the β as it is since it already lies in the unit norm ball. The maximum with 0 essentially denotes the projection of any vector within the unit norm ball to the first quadrant of that ball only.

E.C Proof of the Theorems

As mentioned in the paper the optimization proposed by us can be written in the same format as [42]

$$\underset{M\succeq 0}{\text{minimize}} \;\; L_T(M) + \lambda\|M - M_S\|_F^2 \tag{E.22}$$

where $M_S = \sum_{j=1}^{N} \beta_j M_j$ and

$$L_T(M) = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \mu^\star \big(b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij}\big) \tag{E.23}$$

Theorem 1. *For the convex and k-Lipschitz loss defined in* (E.23) *the average bound can be expressed as*

$$\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}^n}[L_{\mathcal{D}_\mathcal{T}}(M^\star)] \leq L_{\mathcal{D}_\mathcal{T}}(\widehat{M_S}) + \frac{8k^2}{\lambda n}, \tag{E.24}$$

where n is the number of target labeled example, $M^\star$ is the optimal metric computed from Algorithm 1, $\widehat{M_S}$ is the average of all source metrics defined as $\frac{\sum_{j=1}^{N} M_j}{N}$, $\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}^n}[L_{\mathcal{D}_\mathcal{T}}(M^\star)]$ is the expected loss by $M^\star$ computed over distribution $\mathcal{D}_\mathcal{T}$ and $L_{\mathcal{D}_\mathcal{T}}(\widehat{M_S})$ is the loss of average of source metrics computed over $\mathcal{D}_\mathcal{T}$.

Proof. If there is a single source metric is available for transfer , the proof has been shown in [42]. In case of multiple metric for any fixed β, we can directly replace M_S by $\sum_{j=1}^{N} \beta_j M_j$ in the **Theorem 2** in [42] to get,

$$\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}^n}[L_{\mathcal{D}_\mathcal{T}}(M^\star)] \leq L_{\mathcal{D}_\mathcal{T}}\big(\sum_{j=1}^{N} \beta_j M_j\big) + \frac{8k^2}{\lambda n} \tag{E.25}$$

which is true $\forall \beta \in \mathcal{C}_3$. Where,

$$\beta = \begin{bmatrix}\beta_1 & \beta_2 & \dots & \beta_N\end{bmatrix}^\top \in \mathbb{R}^N \tag{E.26}$$

Clearly without loss of generality we can write $\beta = \beta'$ where,

$$\beta' = \begin{bmatrix}\frac{1}{N} & \frac{1}{N} & \dots & \frac{1}{N}\end{bmatrix}^\top \in \mathcal{C}_3 \tag{E.27}$$

since, $\beta' \geq 0$ and $\|\beta'\|_2 = \frac{1}{\sqrt{N}} \leq 1$. So, plugging β' in equation (E.25) we get equation (E.24), which completes the proof. □

Theorem 2. *With probability $(1-\delta)$, for any metric M learned from Algorithm 1 we have,*

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}(\frac{1}{n}) + \left(\sqrt{\frac{L_T(\sum_{j=1}^{N} \beta_j M_j)}{\lambda}} + \quad \|\sum_{j=1}^{N} \beta_j M_j\|_F\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{E.28}$$

where $L_{\mathcal{D}_\mathcal{T}}(M)$ is the loss over the original target distribution (true risk), $L_T(M)$ is the loss over the existing target data (empirical risk), and n is the number of target samples.

Proof. In [42], $L_T(M)$ is defined as,

$$L_T(M) = \frac{1}{n^2} \sum_{(z_i,z_j)\in T} l(M, z_i, z_j) \tag{E.29}$$

□

The authors in [42] have used a specific loss for analysis,

$$l(M, z_i, z_j) = [yy'((z_i - z_j)^\top M(z_i - z_j) - \gamma_{yy'})]_+ \tag{E.30}$$

For our case,

$$\begin{aligned} L_T(M) &= \frac{1}{n_s} \sum_{(i,j)\in S} z_{ij}^\top M z_{ij} + \mu^\star \big(b - \frac{1}{n_d} \sum_{(i,j)\in D} z_{ij}^\top M z_{ij}\big) \\ &= \frac{1}{(n_s+n_d)} \frac{(n_s+n_d)}{n_s} \sum_{(i,j)\in S} z_{ij}^\top M z_{ij} \\ &+ \frac{\mu^\star b(n_s+n_d)}{(n_s+n_d)} - \frac{\mu^\star(n_s+n_d)}{n_d} \cdot \frac{1}{(n_s+n_d)} \sum_{(i,j)\in D} z_{ij}^\top M z_{ij} \\ &= \frac{1}{n^2} \sum_{(i,j)\in T} (\zeta_{ij}(z_i - z_j)^\top M(z_i - z_j) + \gamma) \end{aligned} \tag{E.31}$$

In our case we took similar and dissimilar pairs in equal number. So, for our case $n_s = n_d = \frac{n^2}{2}$ which implies $(n_s + n_d) = n^2$. Also, $\zeta_{ij} = (1 + \frac{n_d}{n_s}) = 2$ if $(i,j) \in S$ and $\zeta_{ij} = -\mu^\star(1 + \frac{n_s}{n_d}) = -2\mu^\star$ if $(i,j) \in D$ are soft labels. Also $\gamma = \mu^\star b(n_s + n_d) = \mu^\star b n^2$. so for our case,

$$l(M, z_i, z_j) = (\zeta_{ij}(z_i - z_j)^\top M(z_i - z_j) + \gamma) \tag{E.32}$$

Also unlike [42] our source metric is defined as $M_S = \sum_{j=1}^N \beta_j M_j$. With the loss in equation (E.32) if we follow the exact same steps as in proof of the **Lemma 2** of [42] then we will end up with the fact that our proposed loss is (σ, m) admissible with $m = 2(1+\mu^\star)\max_{x,x'} \|x - x'\|_2^2 \Bigg(\sqrt{\frac{L_T(\sum_{j=1}^N \beta_j M_j)}{\lambda}} + \| \sum_{j=1}^N \beta_j M_j \|_F \Bigg)$ and $\sigma = 0$.

Now putting these values of σ and m in the equation of inequality of **Theorem 4** of [42] which is,

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}\big(\frac{1}{n}\big) + (4\sigma + 2m + c)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{E.33}$$

and ignoring c and the constant factor which are not functions of source metrics or their weights we conclude our proof.

E.C.1 Finding lipschitz constant for our loss

Goal: Our goal is to show the k in equation (E.24) has a finite value. According to the definition the loss $l(M, x, x')$ is k-lipschitz with respect to its first argument if for any pair of matrices M and M' and pair of samples x and x' we have the inequality as follows for a finite non-negative k ($0 \leq k < \infty$)

$$|l(M, x, x') - l(M', x, x')| \leq k\|M - M'\|_F \tag{E.34}$$

Lemma 3. *The loss defined in equation* (E.32) *is k-lipschitz with* $k = 2\max(1, \mu^\star) \max_{x,x'} \|x - x'\|_2^2$

Proof.

$$\begin{aligned}
|l(M, x_i, x_j) - l(M', x_i, x_j)| &\leq |(\zeta_{ij}(x_i - x_j)^\top M(x_i - x_j) + \gamma) - (\zeta_{ij}(x_i - x_j)^\top M'(x_i - x_j) + \gamma)| \\
&\leq |\zeta_{ij}(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq \max\left(|\zeta_{ij}|\right) |(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq \max(2, 2\mu^\star) |(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq 2\max(1, \mu^\star) \|x_i - x_j\|_2^2 \|M - M'\|_F \\
&\leq 2\max(1, \mu^\star) \max_{x,x'} \|x - x'\|_2^2 \|M - M'\|_F
\end{aligned} \tag{E.35}$$

Comparing this inequality with eq. (E.34) we get
$k = 2\max(1, \mu^\star) \max_{x,x'} \|x - x'\|_2^2$, which is clearly non-negative and finite. □

E.D On-boarding a Single New Camera

This section covers the camera wise experimental results of on-boarding a single new camera (See Figure (E.7,E.9,E.10,E.11). We show for each dataset the camera wise CMC curves that are averaged to a single CMC curve in the main paper. We also showed the comparison of GFK based methods in their original setting where source data is used during target adaptation in WARD dataset (See Figure E.8).

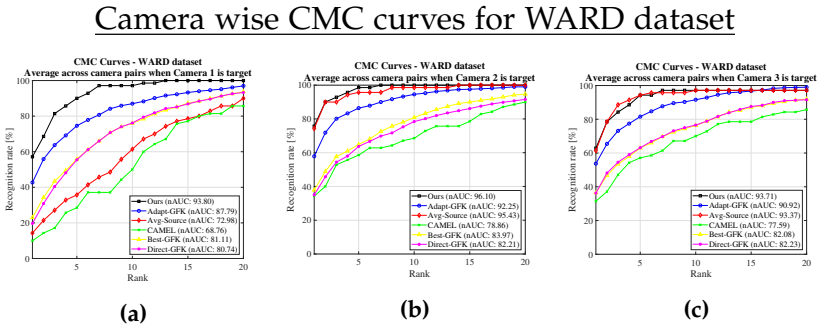


Fig. E.7: CMC curves for WARD [43] with 3 cameras. In this experiment each camera is shown as target while other two cameras served as source. The percentage label of new persons between the new target camera and the existing source cameras is taken to be 20% in this case. The most competitive method here is Adapt-GFK which is outperformed by our method in nAUC with margins 6%, 3.5% and 2.79% for camera 1,2 and 3 as target (plot a, b and c) respectively. In this case Adapt-GFK is calculated using the GFK matrix calculated by only using the limited labelled target data after the installation of new camera. Moreover for camera 1 as target (plot (a)) our method outperforms Adapt-GFK by a large rank-1 margin of almost 16%. Notable thing in this case is that there is only one source metric available for this dataset which is also handled by our multiple source metric transfer algorithm efficiently. Our method significantly outperform the semisupervised method CAMEL for all the plots which shows the strength of our method when a little target labeled data availabe. Also, our method outperforms Avg-Source for all the plots which is a proof of implication of Theorem 1.

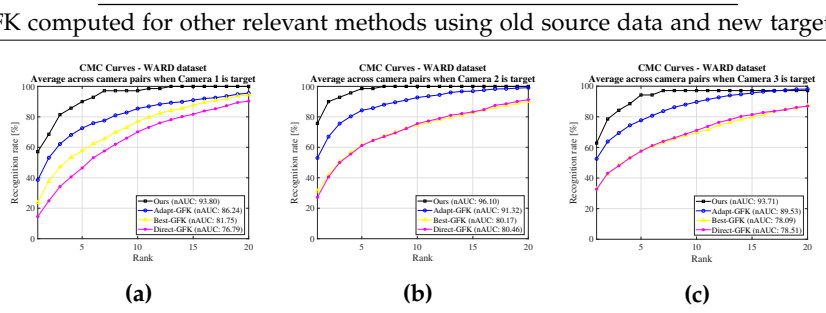


Fig. E.8: The setting in this case is exactly same as the setting of Figure E.7. However this experiment is done only to compare our method with GFK methods in the original settings [12] where the assumption was of the availability of source data. In this case GFK is calculated using the old source data as well as new limited target data. Our method significantly outperforms all the GFK based methods in this case also. It proves that even if our method does not use source data, it still outperforms the doamin adaptation methods which uses source data.

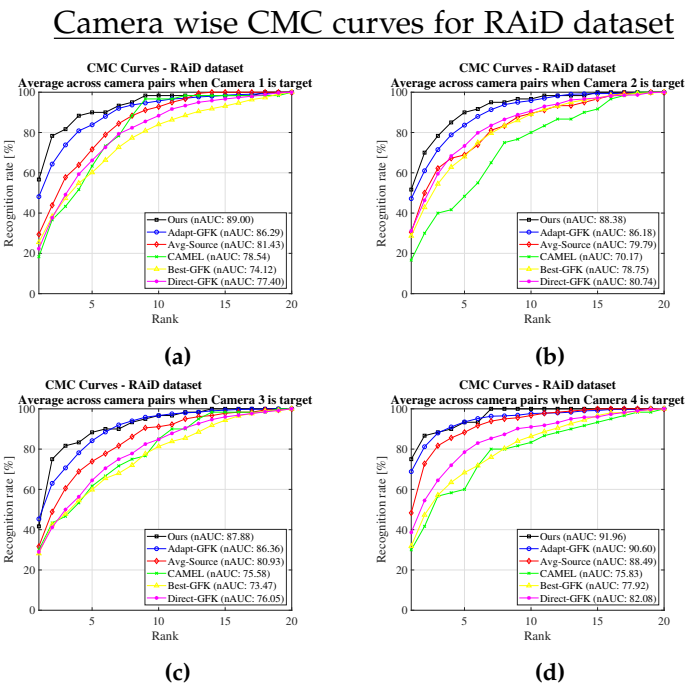


Fig. E.9: In this experiment RAiD dataset with 4 cameras [44] is used. Each of the camera has been set as target while rest of the 3 cameras with 3 pairwise metrics served as source metrics. plot (a,b,c,d) are generated from camera 1,2,3 and 4 as target target camera. The most competitive method here is Adapt-GFK which is outperformed by our method in nAUC with margins 2.71%, 2.2%, 1.52% and 1.36% for camera 1,2,3 and 4 as target respectively. Moreover for camera 1 as target (plot (a)) and camera 4 as target (plot (d)) our method outperforms Adapt-GFK by a rank-1 margin of almost 7% and 5% respectively. Also for each of the cameras our method outperforms Avg-source significantly both in rank-1 and nAUC which proves the Theorem 1. Moreover, for all the cases our method outperforms CAMEL significantly (Like in camera 4 rank-1 margin is almost 36%) which is equivalent to fully supervised learning with limited labels with no transfer from any sources.

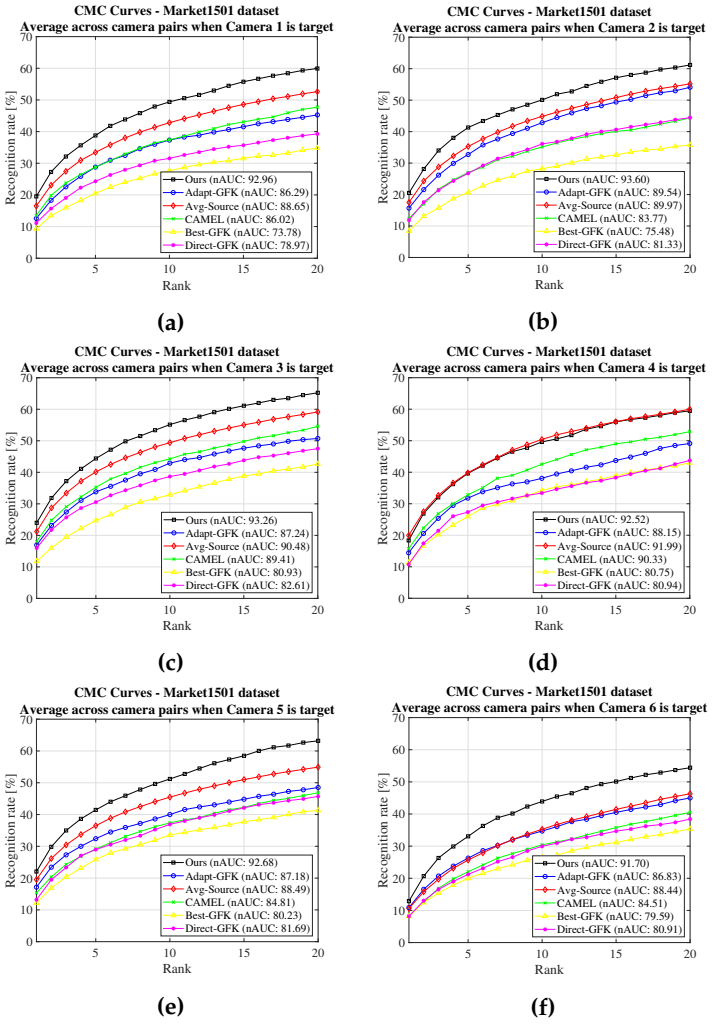


Fig. E.10: In this single camera insertion experiment Market1501 [45] dataset is used. In plots (a,b,c,d,e and f) cmc curves are shown for camera 1,2,3,4,5 and 6 as target respectively. Only 10% of the available data is used between each target-source pairs. Our method outperforms Adapt-GFK which was the most competitive one in case of RAiD and WARD by 6.67%,4.06%,6.02%,4.37%,5.5%,4.87% in nAUC. However, in this case we see that Adapt-GFK has lower accuracy than just the Avg-source, which we outperform in both rank-1 and nAUC for each and every camera as target. Also our method has very high accuracy both in rank-1 and nAUC than CAMEL which is equivalent to no transfer scenario. It is clear that our method gives theoretical guarantee that it would not perform worse than Avg-source case or no transfer case whereas other method has no guarantee which is depicted in this case where Adapt-GFk performed worse than just the Avg-source.

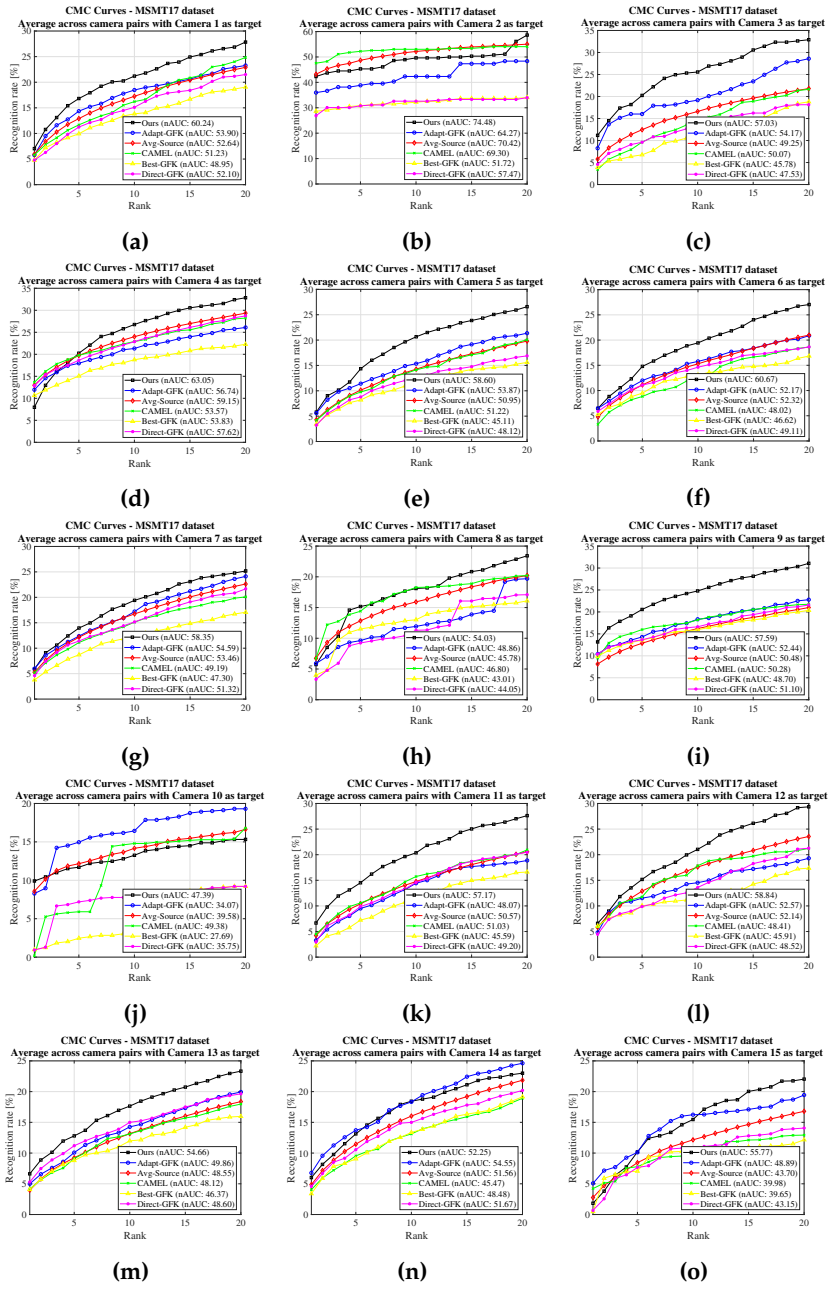
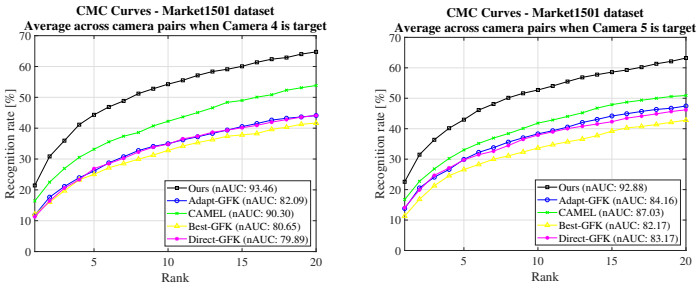


Fig. E.11: Total 15 plots from 15 cameras as target in MSMT dataset are shown. For all cameras our method outperforms other methods in nAUC. While rank-1 performances varied a lot across different cameras, our method on average performs the best as shown in the main paper. Best viewed in color.

E.E On-boarding Multiple New Cameras

This section covers the camera wise experimental results of on-boarding multiple new cameras (See Figure (E.12,E.13,E.14). We show for each experiment the camera wise CMC curves that are averaged to a single CMC curve in the main paper.

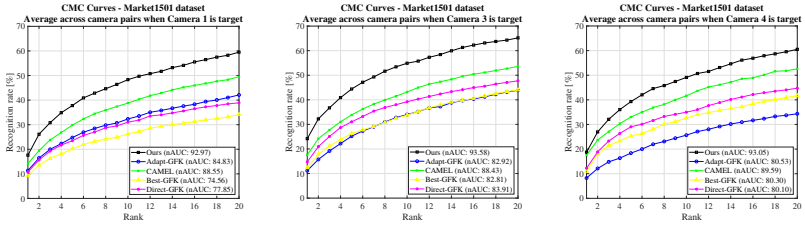
Camera wise CMC curves for Market1501 dataset: parallel addition of 2 cameras



(a) **(b)**

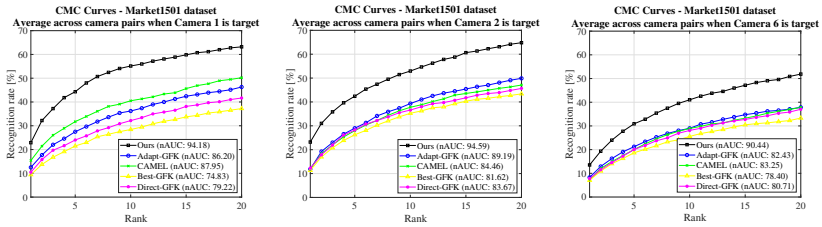
Fig. E.12: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 2 cameras). We effectively set camera 4 and 5 as target and compute 6 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 4 and camera (1,2,3,6) (plot(a)) and also between camera 5 and camera (1,2,3,6) (plot(b)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added in parallel.

Camera wise CMC curves for Market1501 dataset: parallel addition of 3 cameras



(a) **(b)** **(c)**

Fig. E.13: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 3 cameras). We effectively set camera 1,3 and 4 as target and compute 3 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 1 and camera (2,5,6) (plot(a)),camera 3 and camera (2,5,6) (plot(b)) and also between camera 4 and camera (2,5,6) (plot(c)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added in parallel. Best viewed in color.



(a) **(b)** **(c)**

Fig. E.14: In this figure we used Market1501 dataset to show the effect of sequential on-boarding of multiple cameras (In this case 3 cameras). Source cameras are camera 3,4 and 5 which has three source metrics between them. First camera 1 is added to the network and adapted. Accuracy for camera 1 as target is computed between camera 1 and camera (3,4,5) (plot(a)). Then camera 2 is added and adapted. For calculation of camera 2 adaptation accuracy we calculate matching score between camera 2 and camera (1,3,4,5) (plot(b)). In same fashion camera 6 is added afterwards and accuracy is calculated between camera 6 and camera (1,2,3,4,5) (plot(c)). We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added sequentially.

E.F Finetuning with Deep Features

Goal: In this section our goal is to show the performance of our method (See Table E.2 and Figure E.15), if we have access to a deep model trained well using the source data.

Implementation details: This section covers the implementation details of finetuning deep features used in the experiments of Section 5.4 in the main paper. First, we train a ResNet model [51], pretrained on the Imagenet dataset, using the source camera data. We remove the last classification layer and add two fully connected layers; one which embeds average pooled features to size 1024 and another which works as a classifier. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards we fine-tune the model using the new target data and use the new optimized target features along with the source metrics in optimization E.12. The model is trained for 50 epochs using SGD, with a base learning rate of 0.001, which is decreased by a factor 10 after 20 and 40 epochs. We use a batch size of 32 and perform traditional data augmentation, such as cropping and flipping. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards, we fine-tune the model for 30 epochs using the new target data. We fine-tune with a batch size of 32 and a base learning rate is 0.0001 and decreased by a factor 10 after 20 epochs. The new optimized

target features are used along with the source metrics in optimization.

	Single-query		Multi-query	
Method	Top-1	mAP	Top-1	mAP
Euclidean	46.51	40.04	54.40	48.54
Euclidean-ft	51.51	45.52	59.66	54.36
KISSME	45.57	38.42	55.31	48.02
KISSME-ft	49.13	41.77	58.52	51.58
Ours	47.79	41.20	57.57	50.83
Ours-ft	**52.84**	**46.70**	**61.96**	**56.28**

Table E.2: Results for Market1501 when we have a deep model trained using the data of 5 source cameras. We set each camera as target with 25% labeled data in it and show result of average across all the cameras. **Euclidean** denotes the accuracy of target camera if the trained source model is directly used to extract features in target test set. **KISSME** is direct metric learning between new camera and old cameras. *ft* stands for fine tuning. **Euclidean-ft** and **KISSME-ft** is same scheme that is described in the top lines of this section, except for the feature extraction policy. In these methods features are extracted using the fine tuned source model with limited target data. We can see that our proposed algorithm using features from fine-tuned model outperforms all the other accuracies.

CMC curves for Market1501 dataset with Camera 6 as target using deep learned features

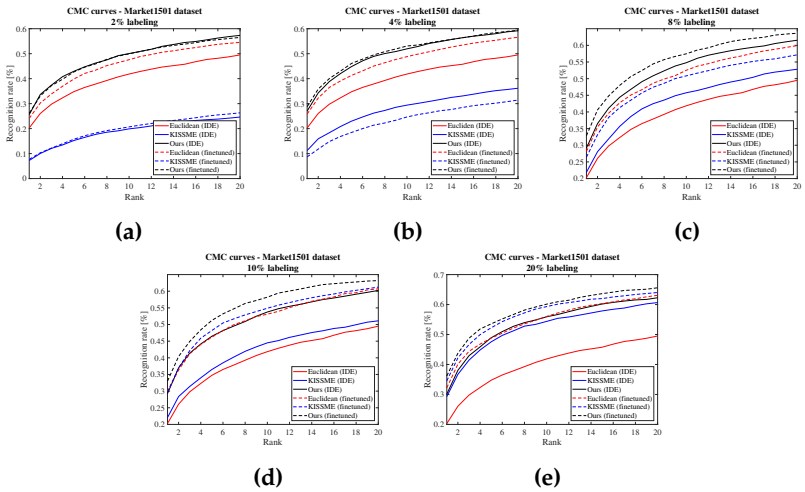


Fig. E.15: These plots show cmc curves for camera 6 of Market1501 dataset using the exact same scheme of Table E.2 but with different percentage labels in the target. We can clearly see that our method outperforms all the other (That is direct euclidean, direct metric learning and even fine tuning with target data). When the percentage label increase then our method with non-finetuned features merges with the direct fine tuning, whereas if we use our method with the finetuned features, it exceeds all the accuracy. This shows the strength of our method even in the presence of deep learned source model.

References

[1] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[2] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[3] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. CVPR*, 2015, pp. 2197–2206.

[4] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Proc. CVPR*, 2016, pp. 1363–1372.

[5] ——, "Hierarchical gaussian descriptors with application to person re-identification," *IEEE transactions on pattern analysis and machine intelligence*, 2019, early Access.

[6] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. CVPR*, 2015, pp. 3908–3916.

[7] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. CVPR*, 2018, pp. 2285–2294.

[8] Y. Sun, Q. Xu, Y. Li, C. Zhang, Y. Li, S. Wang, and J. Sun, "Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification," in *Proc. CVPR*, 2019, pp. 393–402.

[9] C.-P. Tay, S. Roy, and K.-H. Yap, "Aanet: Attribute attention network for person re-identifications," in *Proc. CVPR*, 2019, pp. 7134–7143.

[10] W. Yang, H. Huang, Z. Zhang, X. Chen, K. Huang, and S. Zhang, "Towards rich feature discovery with class activation maps augmentation for person re-identification," in *Proc. CVPR*, 2019, pp. 1389–1398.

[11] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proc. CVPR*, 2019, pp. 2138–2147.

[12] R. Panda, A. Bhuiyan, V. Murino, and A. K. Roy-Chowdhury, "Unsupervised adaptive re-identification in open world dynamic camera networks," in *Proc. CVPR*, 2017, pp. 7054–7063.

[13] ——, "Adaptation of person re-identification models for on-boarding new camera (s)," *Pattern Recognition*, vol. 96, p. 106991, 2019.

[14] J. Lv, W. Chen, Q. Li, and C. Yang, "Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns," in *Proc. CVPR*, 2018, pp. 7948–7956.

[15] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. CVPR*, 2018, pp. 3712–3722.

[16] R. Sun, X. Zhu, C. Wu, C. Huang, J. Shi, and L. Ma, "Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection," in *Proc. CVPR*, 2019, pp. 4360–4369.

[17] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proc. CVPR*, 2019, pp. 5704–5713.

[18] H. Noh, T. Kim, J. Mun, and B. Han, "Transfer learning via unsupervised task discovery for visual question answering," in *Proc. CVPR*, 2019, pp. 8385–8394.

[19] I. Kuzborskij and F. Orabona, "Stability and hypothesis transfer learning," in *Proc. ICML*, 2013, pp. 942–950.

[20] S. S. Du, J. Koushik, A. Singh, and B. Póczos, "Hypothesis transfer learning via transformation functions," in *Proc. NIPS*, 2017, pp. 574–584.

[21] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. CVPR*. IEEE, 2009, pp. 498–505.

[22] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[23] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. CVPR*, 2012, pp. 2288–2295.

[24] X. Yang, M. Wang, and D. Tao, "Person re-identification with metric learning using privileged information," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 791–805, 2017.

[25] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *Proc. ICCV*, 2017, pp. 994–1002.

[26] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proc. CVPR*, 2016, pp. 1249–1258.

[27] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue, "Multi-scale deep learning architectures for person re-identification," in *Proc. ICCV*, 2017, pp. 5399–5408.

[28] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng, "Point to set similarity based deep feature learning for person re-identification," in *Proc. CVPR*, 2017, pp. 3741–3750.

[29] G. Wang, J. Lai, P. Huang, and X. Xie, "Spatial-temporal person re-identification," in *Proc. AAAI*, vol. 33, 2019, pp. 8933–8940.

[30] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proc. CVPR*, 2018, pp. 2275–2284.

[31] H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised person re-identification by soft multilabel learning," in *Proc. CVPR*, 2019, pp. 2148–2157.

[32] Q. Yang, H.-X. Yu, A. Wu, and W.-S. Zheng, "Patch-based discriminative feature learning for unsupervised person re-identification," in *Proc. CVPR*, 2019, pp. 3633–3642.

[33] Y. Lin, X. Dong, L. Zheng, Y. Yan, and Y. Yang, "A bottom-up clustering approach to unsupervised person re-identification," in *Proc. AAAI*, vol. 33, 2019, pp. 8738–8745.

[34] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *Proc. CVPR*, 2018, pp. 5177–5186.

[35] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 4, p. 83, 2018.

[36] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, and N. Zheng, "Semi-supervised person re-identification using multi-view clustering," *Pattern Recognition*, vol. 88, pp. 285–297, 2019.

[37] A. Wu, W.-S. Zheng, X. Guo, and J.-H. Lai, "Distilled person re-identification: Towards a more scalable system," in *Proc. CVPR*, 2019, pp. 1187–1196.

[38] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proc. ACM MM*. ACM, 2007, pp. 188–197.

[39] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. NIPS*, 2009, pp. 1041–1048.

[40] F. Orabona, C. Castellini, B. Caputo, A. E. Fiorilla, and G. Sandini, "Model adaptation with least-squares svm for adaptive hand prosthetics," in *Proc. Robotics and Automation*. IEEE, 2009, pp. 2897–2903.

[41] Y.-X. Wang and M. Hebert, "Learning by transferring from unsupervised universal sources," in *Proc. AAAI*, 2016, pp. 2187–2193.

[42] M. Perrot and A. Habrard, "A theoretical analysis of metric hypothesis transfer learning," in *Proc. ICML*, 2015, pp. 1708–1717.

[43] N. Martinel and C. Micheloni, "Re-identify people in wide area camera network," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 31–36.

[44] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network," in *Proc. ECCV*. Springer, 2014, pp. 330–345.

[45] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. ICCV*, 2015, pp. 1116–1124.

[46] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person trasfer gan to bridge domain gap for person re-identification," in *Proc. CVPR*, 2018, pp. 79–88.

[47] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[48] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*. Springer, 2011, pp. 91–102.

[49] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning." in *Proc. ACCV*. Springer, 2012, pp. 31–44.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[52] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. CVPR*. IEEE, 2012, pp. 2066–2073.

[53] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

Paper F

One-to-One Person Re-identification for Queue Time Estimation

Aske R. Lejbølle, Benjamin Krogh, Kamal Nasrollahi, and Thomas B. Moeslund

This paper is ongoing work, 2019

The layout has been revised.

1 Introduction

The task of matching characteristics, i.e., features, from persons captured across non-overlapping cameras in a camera network is also known as person re-identification (re-id). Person re-id is often linked to forensics where an operator inputs an image of a person (probe) to a system, which is matched against a database of known persons (gallery), and returns a list returned of the most likely matches. In this case, the re-id problem is defined as an image retrieval problem [1–3], and we are satisfied if the correct match is within the, e.g., 10 most likely matches 100 % of the time. Furthermore, this case often consider a close-world setting where we are certain that the same persons have appeared in all cameras. On the other hand, tasks exist, such as trajectory tracking of multiple persons across a large camera network [4, 5], where re-id is considered as a verification task and the probe captured by one camera is matched one-to-one against all persons gallery to verify which that match. In this setting, we do not know which cameras in the camera network have captured the person of interest, therefore, we cannot simply consider a list of likely matches. This is also known as an open-world setting [6]. Finally, we might have a closed-world setting where we do not have an operator to observe a returned list, rather, we wish to find the likely matches to a given set of probes that are actually correct. In this case, we might also consider additional contextual information, such as the number of times a certain person in the gallery can be matched to a given probe.

This work aims to find a set of correctly matched persons from a gallery in a close-world setting using additional contextual information. More specifically, we consider a queue in an airport, where passengers captured by an overhead camera at the exit is matched against those captured by an overhead camera at the entrance, to find the time that passengers spent in the queue, as shown in Figure F.1. Between the entrance and exit, the queue might temporarily split up, thus, while we do know that the same passengers appeared at both entrance and exit, we cannot naively assume a first-in-first-out scenario to measure queue times of each passenger. Instead, we slack the assumption and weight the likelihood of persons to a given probe based on the order of which they entered the queue. Additionally, we make a valid assumption that each person in the gallery can be matched against only a single probe.

We propose to solve this problem as an optimization problem, that is, we wish to reduce the total matching cost between the persons in the gallery and a set of probes. Since in re-id, the likelihood of a match is defined by a distance of some pre-defined metric, minimizing the total cost corresponds to minimizing the total distance between the probes and gallery. We propose to apply the Hungarian algorithm [7], to increase the re-id precision

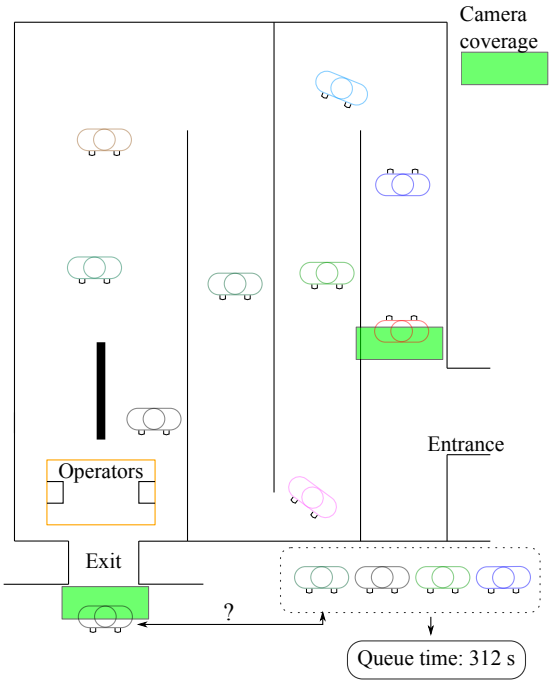


Fig. F.1: Principle of re-id in a queue. Passengers follow a pre-defined maze and are captured by cameras both and the entrance and exit. Once captured by the exit camera, features are matched against all passengers captured by the entrance camera to find a match and output a corresponding queue time.

by matching each gallery to only a single probe with a minimal total cost. The Hungarian algorithm has in re-id previously been used to minimize the cost of matching image patches across probe and gallery [8–10], but has not yet been applied in a post processing step to increase the re-id precision. To summarize, we provide the following contributions:

- We propose the use of person re-id to measure queue times in an airport. To our knowledge, this is the first such work is presented.
- We propose a post processing step to increase the precision of re-id using the Hungarian algorithm to minimize the total distance between probes and gallery.
- Through experiments, we show that re-id based queue time measurements can produce median queue times that are close to ground truth.

2 Methodology

An overview of the proposed methodology is shown in Figure F.2. We extract features from both probes and gallery and perform feature matching. The resulting distance matrix is used as input to the assignment algorithm, which assigns gallery persons to each probe based on minimization of the total distance between probes and gallery.

In this work, we consider the use of an existing convolution neural network (CNN) architecture, which was developed to perform re-id from an overhead viewpoint [11]. The CNN is a multimodal architecture, which processes RGB and depth images in parallel using a MobileNetV2 as backbone [12], and fuses modality features in late layers of the network. Furthermore, the network includes soft attention mechanisms [13] to capture and dynamically weight local semantics for each modality before fusing the two modalities. Please see [11] for a more detailed description of the architecture.

We use the CNN to extract features $f_i^p, f_j^g \in S$ from probes and gallery, respectively, where f_i^p is the feature descriptor from the i'th probe, and f_j^g is the feature descriptor from the j'th gallery. Furthermore, $\{f_i^p, f_j^g\} \in R^{128}$, which is defined by the CNN. Using probe and gallery features, we calculate the Euclidean distance $D_E(f_i^p, f_j^g)$ between the i'th probe and j'th gallery features. To also consider the order of which passengers entered the queue, we add an additional distance, based on assigning entrance and exit id's to each passenger. The id's are assigned based on the order of which a passenger entered and exited the queue, respectively. Note that, the same passenger might have different entrance and exit id's due to an intermediate queue split. We define the entrance and exit id's as $enter_j$ and $exit_i$, respectively, and calculate the order distance between the i'th probe and j'th gallery as $D_O(enter_j^g, exit_i^p) = \log(1 + |enter_j^g - exit_i^p|)$. We take the logarithmic value of the distance, since we can have potentially large values compared to Euclidean distances, depending on the number of persons in the dataset.

Next, we apply the Hungarian algorithm to find an optimal set S^* of matches that solve the following optimization problem:

$$\begin{aligned} \mathcal{S}^* = &\arg\min_{\mathcal{S}} \sum_{\{f_i^p, f_j^g, enter_j^g, exit_i^p\} \in \mathcal{S}} D_E(f_i^p, f_j^g) + D_O(enter_j^g, exit_i^p) \\ &\text{Subject to} \quad f_i^p \neq f_m^p, f_j^g \neq f_n^g \quad \forall \{f_i^p, f_j^g\}, \{f_m^p, f_n^g\} \in S \end{aligned} \tag{F.1}$$

where the optimal set of S^* is the one where the sum of distances between features and id's in the set S is minimized and $\{f_i^p, f_j^g\}, \{f_m^p, f_n^g\}$ are matched pairs. The constraint indicate that each of the feature descriptors in a matched pair cannot be matched to any other feature descriptors.

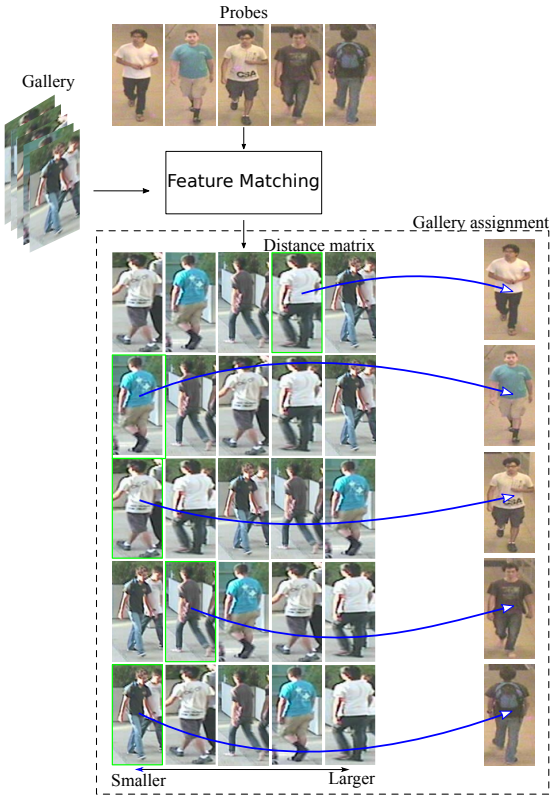


Fig. F.2: Overview of the assignment procedure. We match each probe against all gallery features to produce initial lists based on likelihood, which is defined by distances. Images further to the left have smaller distances to the probes and vice versa, while green boxes indicate true matches. Afterwards, the Hungarian algorithm [7] is applied to assign each gallery to only one of the probes by minimizing the total distance.

3 Experiments

We wish to evaluate the proposed algorithm on an overhead re-id dataset with timestamps available. To our knowledge, only the public dataset of [14] contains timestamps, however, the dataset is captured in an outdoor environment at a university campus, which does not comply with our goal of using timestamps to measure queue times. As a result, we have collected and annotated a new dataset we call *Queue Person Re-identification* (QPR). In the following, the dataset is presented along with the experimental results of our proposed assignment algorithm.

3.1 Dataset

We collect data from an immigration area at an airport to properly evaluate queue time measurements using vision-based person re-identification. Data is collected in the morning using two ZED cameras [15] that are placed overhead at non-overlapping locations. The first camera is placed at the entrance of the immigration area while the second is placed at the exit, thus, capturing the queue times of each passenger. Data is collected in 2k resolution (2208×1080) at 15 frames per second (FPS). We extract RGB images and, similarly to [16], compute disparity maps using semi-global block matching (SGBM) followed by filtering to smoothen the disparity maps. Additionally, disparity maps are converted to depth maps, and a JET color map is applied to create depth images that can be used to train the CNN. We annotate bounding boxes around all persons in the dataset, resulting in 7529 bounding boxes across 116 persons. An overview of the dataset statistics is shown in Figure F.3. A few persons, primarily those that entered the queue late, have more than 200 annotated bounding boxes, nonetheless, the majority of persons in the dataset have [50,100] annotated bounding boxes, as seen in Figure F.3 (b). Furthermore, Figure F.3 (c) shows the ground truth queue times of each passenger based on enter time, starting at zero from the entrance of the first passenger. Naturally, as more passengers enter the queue, the queue time increases. Related to Figure F.3 (a), we see that the number of bounding boxes per person increases as the queue time increases. This is due to the queue eventually reaching the entrance and slowing down, leaving persons within the view of the camera for a longer period of time.

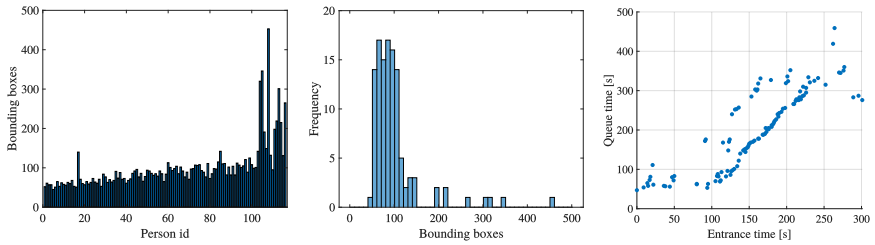


Fig. F.3: (a) Number of bounding boxes per person, (b) frequency of bounding boxes in bins of 10, and (c) ground truth queue times based on time of entrance starting at zero for the first person.

3.2 Implementation Details

As mentioned in section F.2, we apply an existing CNN architecture [11]. The QPR dataset is split between 26 persons that are used for training and 90 that

are used for testing. The network is trained by using a combination of cross-entropy and triplet loss with soft margin from hard positives and negatives [17], which has been a popular approach to train CNNs in re-id [18–21]. We use Adam [22] to optimize the weights of the CNN, with a base learning rate of 0.0001, which is reduced by a factor of 0.9 every 100 epochs, and decay values of 0.9 and 0.999, respectively. Finally, we train the network using a batch size of 32 until convergence, but up to 1000 epochs. For each batch, we randomly sample four images from eight randomly sampled persons.

Upon testing, we extract features from all images of each person, and perform average pooling on the features to create a single feature descriptor for each person in the two views. We match feature descriptors between the two views using Euclidean distance, and correspondingly, calculate the order distances between probes and gallery. The output distance matrices are then used as input to our assignment algorithm. We report precision and recall values, and compare to naively assigning using the most likely match.

3.3 Experimental Results

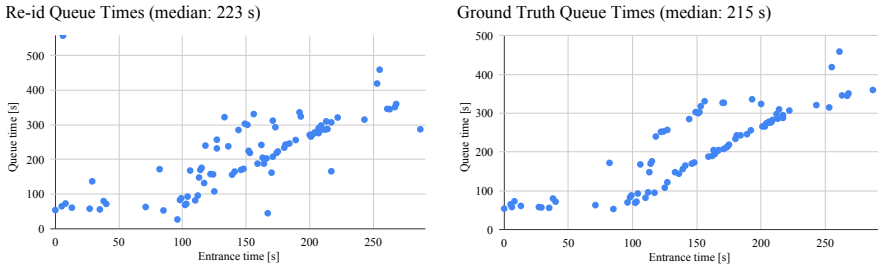
We perform various experiments where we consider different numbers of most likely matches when assigning gallery to probes, that is, we consider the top-k matches in our assignment. Naturally, in the case that k is less than the number of persons in the gallery, not all persons may necessarily be assigned to a probe if multiple, initially, are assigned the same probes. The results are shown in Table F.1. If we simply assign the most likely match, we achieve a precision and recall of 50 %, which is similar to the rank-1 accuracy of the cumulative matching characteristic (CMC) curve. If we apply our assignment algorithm and just consider the most likely match, we see a large increase in precision, however, at the cost of a lower recall. Nonetheless, considering the six most likely matches, both the precision and recall increase to values that are 22 % and 16 % higher, respectively, compared to the naive approach. Finally, we see that the precision and recall converge if 10 or more most likely matches are considered.

	Naïve	Top-1	Top-3	Top-6	Top-10	Top-20	Top-50	Top-90
Precision [%]	50	71	65	76	67	66	66	66
Recall [%]	50	46	57	70	67	66	66	66

Table F.1: Precision and recall of our proposed assignment algorithm by considering 1, 3, 6, 10, 20, 50 and 90 most likely matches. As reference, we compare to naively assigning by taking the most similar match.

We use the results from Table F.1 to calculate median queue times and plot the measured queue times in relation to queue entrance time in Figure F.4. For comparison, we also plot the ground truth queue times. As seen in the

two plots, similar tendencies are followed with slightly more scatter in case of the re-id based queue times. The differences in queue times results in a median difference of 8 seconds, which is a deviation of only 3.60 %. Depending on the requirement of the airport, this may or may not be an acceptable difference.



(a) **(b)**

Fig. F.4: (a) Measured queue times from re-id and (b) ground truth queue times.

4 Conclusion

In this work, we propose person re-identification to measure queue times in an airport. Different from common re-id tasks, we do not wish to return a list likely matches, rather, perform one-to-one matching based on the ranked lists. We propose a post processing step, which applies the Hungarian algorithm to minimize the total distance between a gallery and a set of probes, by assigning each person in the gallery to only one probe, and vise versa.

To evaluate the proposed algorithm, we have collected a novel overhead person re-id dataset from two non-overlapping cameras in an airport, where timestamps from each camera are available to measure queue times based on re-id results. Through experimental results, we show that the proposed assignment algorithm can increase re-id precision by up to 22 %, which results in a median queue time that deviate 3.60 % from ground truth.

5 Future Work

This work provides preliminary experimental results to measure queue times based on re-id. While we have shown median queue times that are close to ground truth, questions remain to be answered, including:

- How well does vision-based re-id compare to current solutions for measuring queue times?

- What is the processing time of measuring queue times using vision-based re-id? Can this be reduced and at what cost?
- What is the required precision of the re-id system in order to produce median queue times that are close enough to the ground truth median?
- How well does the use of Hungarian algorithm compare to related methods, such as those that consider the assignment problem as a min-cost flow problem [23]?
- Is it possible to further increase re-id precision if an additional distance threshold is considered upon assigning gallery to probes?

Finally, experiments have to be run across multiple random train/test splits as is usually done in evaluation of re-id datasets [24–26].

References

[1] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. CVPR*. IEEE, 2015, pp. 1741–1750.

[2] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[3] Q. Leng, M. Ye, and Q. Tian, "A survey of open-world person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[4] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proc. CVPR*, 2016, pp. 1420–1429.

[5] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. CVPR*, 2017, pp. 3539–3548.

[6] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person re-identification*, 1st ed., ser. Advances in Computer Vision and Pattern Recognition, S. Gong, M. Cristani, S. Yan, and C. C. Loy, Eds. London: Springer, 2014, vol. 1, ch. 1, pp. 1–20.

[7] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[8] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, and J. Wang, "Person re-identification with correspondence structure learning," in *Proc. ICCV*, 2015, pp. 3200–3208.

[9] W. Lin, Y. Shen, J. Yan, M. Xu, J. Wu, J. Wang, and K. Lu, "Learning correspondence structures for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2438–2453, 2017.

[10] S. Bak and P. Carr, "One-shot metric learning for person re-identification," in *Proc. CVPR*, 2017, pp. 2990–2999.

[11] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] M. Zheng, S. Karanam, and R. J. Radke, "Rpifield: A new dataset for temporally evaluating person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 1893–1895.

[15] Stereolabs. (2017) Zed - depth sensing and camera tracking. https://www.stereolabs.com/zed/. Stereolabs. Accessed: November 28, 2019.

[16] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[17] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[18] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *Proc. CVPR*, 2018, pp. 4099–4108.

[19] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proc. CVPR*, 2018, pp. 1179–1188.

[20] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual attention matching network for context-aware feature sequence based person re-identification," in *Proc. CVPR*, 2018, pp. 5363–5372.

[21] J. Song, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Generalizable person re-identification by domain-invariant mapping network," in *Proc. CVPR*, 2019, pp. 719–728.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[23] L. R. Ford Jr and D. R. Fulkerson, "Flows in networks," RAND Corporation, Tech. Rep., 1962.

[24] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. ECCV*. Springer, 2008, pp. 262–275.

[25] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. CVPR*, 2014, pp. 152–159.

[26] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

Paper G

Evaluation of Edge Platforms for Deep Learning in Computer Vision

Aske R. Lejbølle, Christoffer Bøgelund Rasmussen, Kamal Nasrollahi, and Thomas B. Moeslund

Submitted to the journal of Neural Computing and Applications, 2019

The layout has been revised.

Abstract

In recent years, companies, such as Intel and Google, have brought onto the market small low-power platforms that can be used to deploy and run inference of Deep Neural Networks at a low cost. These platforms can process data at the edge, such as images from a camera, to avoid transfer of large amount of data across a network. To determine which platform to use for a specific task, practitioners usually compare parameters, such as inference time and power consumption. However, to provide a better incentive on platform selection based on requirements, it is important to also consider the platform price. In this paper, we explore platform/model trade-offs, by providing benchmarks of state-of-the-art platforms within three common computer vision tasks; classification, detection and segmentation. By also considering the price of each platform, we provide a comparison of price versus inference time, to aid quick decision making in regard to platform and model selection. Finally, by analysing the operation allocation of models for each platform, we have identified operations that should be optimised, based on platform/model selection.

1 Introduction

Within the years that followed 2012, researchers were focused on developing Deep Neural Networks (DNNs) that were accurate and generalised well. Object classification, wherein the big breakthrough had happened, each year saw a decrease in top-1 error on the large ImageNet dataset [1, 2]. As other computer vision (CV) tasks gained more interest, such as object detection and semantic segmentation, accuracy on benchmark datasets would continue to increase each year [3, 4]. However, recently, focus has shifted towards more practical usage of DNNs. Today, more effort is being put into lowering the network complexity while maintaining a high accuracy. Novel architectures are developed that contain fewer parameters [5–7], and larger networks are quantified to speed up inference. The most common datatype for DNNs today is the 32-bit floating point (FP32), however, using quantification techniques, networks can be converted to operate on 16-bit floating point (FP16), or even 8-bit integers, with almost no loss in precision [8, 9].

Following the trend within academia of developing DNNs, more companies are developing hardware to run these networks. This hardware, furthermore, should be able to process incoming data with low latency. Several cloud solutions, offered by big companies, such as Google's Cloud ML Engine [10], Microsoft's Azure [11] and Amazon's Amazon Web Services (AWS) [12], have emerged that can be used to train and run models online. Furthermore, Internet of Things (IoT) have resulted in products that require smaller and cheaper computers, which can be used to run already trained models at the edge. As a result of this demand, companies like Intel and

NVIDIA have brought onto the market edge platforms that can be used to deploy and run network inference at limited costs [13, 14]. These platforms can, for example, be integrated with a camera to process data directly at the source. In the last few years, several minor and large companies have brought onto the market their own edge platforms, combined with software packages to optimise pre-trained models before deployment. These platforms are able to run models within a variety of CV tasks, including object classification and detection.

In this work, we evaluate edge platforms on common CV tasks, including object classification, object detection and semantic segmentation. To this end, we evaluate DNN models of different precision and complexity within each task, to show and compare inference timings between high-precision complex models and medium-precision/simple models when the batch size is varied. For better comparisons between platforms, we also evaluate a high-end GPU and use it as reference. Furthermore, we calculate the number frames per second (FPS) based on the inference timings, and include the retail price of each platform to calculate an FPS cost. The FPS cost can be seen as a measure of how cost effective a certain platform/model combination is, using a certain batch size. Additionally, comparing retail price and FPS, we propose a framework which aid the optimal platform/model selection, depending on budget and speed requirements. Finally, we compare the distribution of DNN operations across platforms and models, and compare this to the FPS costs, to identify which parts of a DNN on certain platforms that correspond to higher FPS costs.

Previous works have studied models of different complexities [15–17], however, these publications often aim to provide an analysis of the speed/accuracy trade-off between models. On the other hand, work has also been published that evaluates and compares different edge platforms [18, 19], but these works do not take into consideration the price of different platforms. By including the price of the platforms, we are able to provide a simple and extensive overview of the FPS cost, which can be used by companies to select the optimal platform/model combination depending on their requirements and resources.

The remaining of the paper is structured as follows. Section G.2 provides an overview of most common classification, detection and segmentation models and surveys hereof that focus on the speed/accuracy trade-off. Additionally, previous work on platform benchmarks will also be outlined. Section G.3 provides an overview of the selected models and platforms for our evaluation while also the procedure of the evaluation will be described. In Section G.4, the results of the evaluation are presented and analysed while the results are discussed in Section G.5. Finally, a conclusion is presented in Section G.6.

2 Related Work

In the following, the most well known architectures within each of the selected CV tasks will be introduced. Since we focus on a comparison between models, details will be neglected in favour of a discussion on related work that, in one way or another, compare models.

2.1 Object Classification

Following AlexNet [20], which consisted of regular convolution, activation, max pooling and fully connected (FC) layers, Convolutional Neural Networks (CNNs) quickly became more sophisticated. By the introduction of skip connections in ResNet [21], Inception modules in GoogLeNet [22] and combinations of the two methods [23], accuracy quickly rose on a variety of classification tasks. Meanwhile, less complex CNNs started to appear, such as the famous MobileNet [24], the extension MobileNetV2 [7] and ShuffleNet [6]. More recently, Zoph et al. [25] proposed using neural architecture search to develop small CNNs that maintain high accuracies.

In the last couple of years, work has been published that compare classification models and their performance. Canziani, Culurciello and Paszke [16] analysed inference time, power consumption and system memory utilisation for models of different complexity, depending on the batch size. The models included AlexNet, VGG, GoogLeNet, variations of ResNet and the Inception nets. Based on all results, a 'top-1 accuracy density' was presented, which indicates the accuracy per million parameters. However, all tests were performed on a single NVIDIA Jetson TX1, and only complex models were considered. Biano et al. [15] extended the work of [16] by including several additional CNNs, while also performing the evaluation on an NVIDIA Titan X GPU, but considered the same parameters. Meanwhile, Velasco-Montero et al. [26] evaluated models of different complexity, implemented in different frameworks, on a low-power Raspberry Pi 3 model B, and considered accuracy, throughput and power consumption to find a subset of optimal model/framework combinations for real-time deployment.

More recently, Almeida et al. [18] conducted an evaluation of several classification models, including those present in [15], but also less complex models. Furthermore, they considered five different platforms, including an edge platform. Similar to [15, 16], they compared inference time and accuracy between models, but rather than having a single plot from all platforms, the comparison was performed per platform to identify differences and similarities between the platforms with respect to the handling of the networks. This was further highlighted by a per platform comparison of time spent on different layer types of a given model. While the work provides insight on how to build up an architecture based on the platform, it does not consider the

cost of using a certain platform.

2.2 Object Detection

With the introduction of Region CNN (R-CNN) by Girshick et al. [27] in 2014, and the extensions of Fast R-CNN [28] and Faster R-CNN [29] the following year, object detection using CNNs accelerated. These networks are based on a two-stage approach, where a Region Proposal Network (RPN) is used to identify class agnostic object proposals, which are then fed to a classifier to predict the presence of an object from a predefined set of classes within each proposal. To speed up detection, one-stage detectors were later developed that combine the two stages into a single network. Most notable are the You Only Look Once (YOLO) by Redmon and Farhadi [30] and the Single Shot Multibox Detector (SSD) by Liu et al. [31]. While one-stage detectors excelled in terms of speed, they fell short in accuracy compared to the two-stage detectors. Only recently, following the proposed RefineDet by Zhang et al. [32] one-stage detectors have started to reach the accuracy of two-stage detectors while maintaining a high speed.

Inference time and accuracy of detectors are mostly dependent on the utilised feature extractor in the object detection pipeline [17]. While most object detectors are evaluated on fast GPUs, such as the NVIDIA Titan X, they are more rarely evaluated on edge platforms. However, by changing the feature extractor to a more simple model, such as MobileNet [24], it is possible to reach low inference timings on these platforms. Huang et al. [17] performed a comparison of different object detectors by changing the feature extractor, to analyse the change in accuracy/speed/memory trade-off in three popular object detectors. The work would serve as a guide to choose the optimal detector/feature extractor combination. Liu et al. [33] presented a more extensive survey of object detectors, where less complex detectors were also considered, such as the SSD using a MobileNet [24] and Light Head R-CNN [34]. However, the survey does not include a speed/accuracy analysis between the presented detectors. To our knowledge, no published work compare speed/accuracy and price across several platforms, including edge platforms.

2.3 Semantic Segmentation

Semantic segmentation is a more challenging task compared to detection and classification, and CNNs on this task are, typically, more processing heavy. Inspired by CNNs for object classification, one of the first segmentation networks, by Long, Shelhamer and Darrell [35], transformed classification networks [20, 22, 36], into Fully Convolutional Networks (FCNs) by outputting spatial maps instead of classification scores. Since then, several segmenta-

tion networks inspired by FCN have been proposed, including SegNet by Badrinarayanan, Kendall and Cipolla [37], and U-Net by Ronneberger, Fischer and Brox [38]. Instead of convolutions to upscale features, the more recent DeepLab and DeepLabV3 by Chen et al. [39, 40] use Atrous Spatial Pyramid Pooling (ASPP) to more effectively segment object at different scales without increasing the complexity.

Few works have been published in benchmarking of semantic segmentation networks. Guo et al. [41] provide an overview of different architectures with the purpose of identifying strengths, weaknesses, and challenges of current work. A more general survey by Garcia-Garcia et al. [3] was published that presents the key ideas behind segmentation networks and provide an overview of previous proposed architectures with focus on, among other things, accuracy and efficiency. While they provide a comprehensive overview, they do not directly compare models.

2.4 Platform Benchmarks

Only few works compare performance of models of different complexity across difference platforms. Trindade et al. [42] evaluated two popular frameworks, Caffe [43] and TensorFlow [44], using AlexNet [20] and GoogLeNet [22], and compared performance, with respect to training time, between a GPU and NUMA CPU. A more extensive evaluation of frameworks, such as TensorFlow and Caffe2 [45], was presented by Zhang, Wang and Shi [46] who performed the evaluation on different platforms, including the NVIDIA Jetson TX2 and a Nexus 6P. Furthermore, they considered two different classification models, the more complex AlexNet [20] and the simple SqueezeNet [5]. They evaluated inference time, memory footprint and energy consumption. Blouw et al. [19] measured inference time and energy consumption across different platforms, including the TX1, Intel Neural Compute Stick (NCS) and their own Loihi chip, with respect to batch size, and analysed the speed and energy cost per inference as a function of the network size. However, they only evaluated platforms on a single custom architecture. Finally, Pena et al. [47] focused on low-power devices, by evaluating object classification models and frameworks across an Intel NCS, Raspberry Pi 3 model B and Intel Joule 570X, with respect to inference time and power consumption.

To our knowledge, only a single previous publication has compared different platforms across different tasks, which is the aim this work. Ignatov et al. [48] considered mobile platforms containing chips that are manufactured by some of the major chipset companies, including Qualcomm, HiSilicon, MediaTek, Samsung, Google and Arm. The chips were evaluated in nine tests, including two image recognition tests using MobileNet and Inception V3, respectively, and a memory limitation test to identify the maximum allowed image size for inference before running out of memory. Instead, we

perform evaluation of edge platforms across different common CV tasks, consider the retail price of the platforms, and analyse the consequence of DNN operations across platforms.

3 Platform Evaluation

This section presents an overview of our methodology for evaluating the edge platforms. Specifically, we present the evaluation procedure to ensure comparable results between platforms, choice of deep learning framework, and overview of the models and platforms.

3.1 Model Overview

The choices for method and models are based upon differences in the complexity of feature extractors dependent on the difficulty of a given task. Therefore, for each of the three tasks covered in this survey, models at up to three different levels of complexity are evaluated. For all tasks, complexity is defined as the number of Giga Floating Point Operations (GLOPS).

For simplicity, we adopt pre-trained networks available in the official TensorFlow [44] framework. A short description of the tasks and their model architectures will also be described.

Classification

This task is arguably the simplest of the object recognition tasks chosen in this work, and requires only a feature extractor in the form of a number of convolutional layers and, typically, one or more FC layers for classification. We adopt MobileNetV1 [24] as the small, ResNet50 [21] as a medium, and InceptionResNetV2 [23] as the larger more complex network.

MobileNetV1 is the first version of Google's efficient and small classification networks that was designed with mobile and embedded systems in mind. MobileNetV1 introduced depthwise separable convolutions to significantly reduce model size and complexity. Depthwise separable convolutions replace standard convolutions by factorising the operation into a depthwise convolution followed by a pointwise convolution. The depthwise convolution convolves a single filter to each layer input, after which, the pointwise convolution combines the outputs from the depthwise layer with a 1×1 convolution. The MobileNetV1 architecture contains a total of 28 layers when the depthwise and pointwise layers are counted separately. This includes the final FC and softmax layers for classification. MobileNetV1 also introduced two parameters to further tune the model size and complexity. The first parameter, a width multiplier α, scales the number of input layers uniformly throughout the network. $\alpha \, \epsilon \, (0, 1]$, where typical values are 1, 0.75, 0.5 and

0.25. In this work, we evaluate the MobileNetV1 trained with $\alpha = 1$. The second parameter, ρ, known as the resolution multiplier, reduces the representational power of the network by scaling the input image and all consequent layers in the network. $\rho \, \epsilon \, (0, 1]$, where for classification it is typically set such that the height and width is 224, 192, 160 or 128 pixels. In this work, we evaluate the MobileNetV1 model trained with $\rho = 1$ resulting in a resolution of 224×224 pixels.

The medium sized network, ResNet50, in 2015 came forth as a large breakthrough in learning deeper networks. At the time, it won numerous competitions including ImageNet classification and localisation, and COCO detection and segmentation, where it was used as feature extractor. The motivation behind ResNets was that deeper models should be able to learn richer and more abstract features. However, as concluded by the authors, training deeper networks stacked with traditional layers is difficult, largely due to a degradation in the training error. To address this degradation, ResNets learn a residual function with reference to the previous layer inputs rather than learning a direct mapping. This, relatively simple reformulation, enabled ResNets to be trained easier at larger depths and resulted in significant accuracy increases. The residual function to be learnt at various layers in the network is defined as $y = F(x, \{W_i\}) + x$, where $(x, \{W_i\})$ is the residual mapping to be learnt and x is the input from the previous layer.

InceptionResNetV2 is a later iteration of the Inception networks originally presented in [22], which also addressed the challenges in training deeper networks. As in the original GoogLeNet [22], the solution was to have wider layers that have filters with multiple sizes whose outputs would be concatenated. A number of improvements have been made between presentation of the first Inception module and the InceptionResNetV2 while keeping the concept of wider layers. This includes factorising convolutions for speed, adding batch normalisation and residual connections inspired from ResNets.

An overview of the classification models described can be seen in Table G.1.

Model	**Year**	**GFLOPS***	**Top-1 [%]**
MobileNetV1 [24]	2017	1.15	70.9
ResNet50 [21]	2015	6.97	75.2
InceptionResNetV2 [23]	2017	26.36	80.4

* As measured in TensorFlow

Table G.1: Overview of classification models. Top-1 accuracy is based on the ImagenNet classification task [49].

Object Detection

For benchmarking object detection networks, we use the SSD [31] with different backbones due to the speed of the networks, which makes them viable for embedded platforms. The distinction between the size of the SSD networks is done by switching the feature extractor which can be used for classification, as described in Section 3.1. First, we provide an overview of the SSD after which we present two networks based upon their feature extraction complexity.

The key increase in speed in SSD, in comparison to two-stage detectors, is due to removing the proposal stage, such as the RPN in Faster R-CNN [29], and by not resampling features in the later stage of the network. SSD follows the structure of many other deep learning based object recognition systems by having a base network that follows an image classification architecture for initial feature extraction. A number of additional layers are added on top of the base network that decrease input size and allows the SSD to make predictions at multiple scales. At each layer, a small kernel traverses the feature map and produces bounding boxes and class scores from a predefined set of boxes. These predefined boxes are similar to those of anchor boxes in Faster R-CNN [29] as they have varying aspect ratios that aim to fit varying object dimensions. However, they are here applied at the multiple scaled layers instead of a single feature layer. The feature map is also discretized to speed up the predictions, as running the kernel of all points in the map would be too slow. Typically sizes for the feature map include 8×8 and 4×4. Finally, Non-Maximum Suppression (NMS) removes overlapping detections in the output.

Table G.2 summarises our choices for the two feature extractors with varying complexity, namely, MobileNetV1 and InceptionV2.

Model	Year	GFLOPS*	mAP [%]
SSD MobileNetV1 [24]	2015	2.49	21
SSD InceptionV2 [17]	2017	9.63	24

* As measured in TensorFlow

Table G.2: Overview of detection models. mAP is based on the COCO detection task [50].

Semantic Segmentation

We adopt DeepLabV3 [40] for evaluating semantic segmentation networks with varying complexity. Naturally, DeepLabV3 is the third iteration of the DeepLab networks, where the original concept of DeepLab [39] was to, amongst other improvements, incorporate 'atrous convolutions' also known

as *dilated convolutions*. The authors introduced the concept as they aimed to address the issues of CNNs excelling at higher-level tasks, such as classification, but less so on more difficult tasks, such as pixel-level segmentation. This issue is argued to be largely due to downsampling in networks through striding and pooling operations, which reduces the memory requirement and is sufficient in creating features for global tasks such as classification, however, discards important finer spatial information. Atrous convolution allows a filter to increase its field-of-view by adding an atrous rate, r. Setting $r > 1$ inserts zeros (holes) between two consecutive weights in a filter, for example, $r = 2$ expands the filter such that every second value is a zero, effectively increasing a 3×3 filter's receptive field to that of a 5×5 while maintaining complexity. In DeepLabV3, the atrous convolutions were extended into an ASPP module. In the module, four sets of atrous convolutions are computed from a feature map, each with their respective atrous rate r. The motivation of ASPP is that the set of convolutions is able to capture multi-scale information in a pyramid of different receptive fields. In addition to the pyramid of convolutions, a global context is captured through an image average pooling operation. The four atrous convolutions and pooling operation are concatenated and a 1×1 convolution is applied to produce an output feature map.

Finally, an overview of model backbone choices for evaluation of DeepLabV3 is shown in Table G.3, which in this case is MobileNetV2 and Xception65.

Model	Year	GFLOPS*	mIOU [%]
DeepLabV3 MobileNetV2 [7]	2018	17.69	75.32
DeepLabV3 Xception65 [40]	2017	354	82.20

* As measured in TensorFlow

Table G.3: Overview of segmentation models. mIOU is based on the VOC 2012 segmentation task [51].

3.2 Platform Overview

This section introduces the platforms evaluated across the various classification, object detection and segmentation models. An overview of some of the key specifications for the platforms can be seen in Table G.4, covering the number of cores, clock frequency, memory, Thermal Design Power (TDP) and price.

Platform	Cores	Clock Freq. (GHz)	Memory (GB)	TDP (W)	Price* ($)
i7-7700K	4	4.2	64	91	300
Intel NCS	12**	0.6	0.5	1	69
Intel NCS 2	16**	0.7	0.5	1	86
NVIDIA GTX 1080	2560***	1.6	8	180	520
NVIDIA Jetson TX2	256***	1.3	8	7.5	570

* Price per 01/01/2019 [52] ** SHAVE cores *** CUDA cores

Table G.4: Overview of evaluated platforms, including the reference GTX 1080.

CPU (i7-7700K)

Within deep learning, the GPU is the most important piece of hardware, however, a CPU-based solution may be necessary due to a number of reasons, such as the price or space restrictions. In this work, we evaluate the Intel i7-7700K CPU, which is part of the Kaby Lake series. The CPU in workstations that include a GPU is more typically used for a number of tasks that do not include matrix computations such as image loading and preprocessing, however, in this work it will be evaluated for deep learning. Additionally, we evaluate the CPU performing inference with 1, 2, and 4 cores.

Intel NCS/NCS2

The Intel NCS is an edge low-power device for performing inference of DNN models. The NCS has the form of a USB stick with dimensions of 72.5 mm $\times$ 27 mm $\times$ 14 mm and must be connected to a host machine via a USB interface [53]. The host machine can be either a Linux, Raspbian or Windows, in this work an Ubuntu 16.04 host machine was used. The NCS is powered by an Intel Movidius Myriad 2 Vision Processing Unit, which has 4 Gbits of LPDDR3 DRAM and 12 shave cores [53]. In order to perform inference on the NCS, the TensorFlow models must be converted into a graph file via the OpenVINO toolkit [54]. We also evaluate the newer Intel NCS2, which follows much of the above but has improved specifications, including an Intel Movidius Myriad X VPU with 4 Gbits of LPDDR4 DRAM and 16 shave cores [13].

NVIDIA Jetson TX2

The NVIDIA Jetson TX2 is the third release in the Jetson edge GPU series. TX2 is, as the Intel NCS and NCS2, designed for inference in low-power scenarios. In this work, we evaluate the developer kit version, however, the TX2 can also be purchased as a standalone module. The TX2 has 256 CUDA cores for deep learning applications, a dual core Denver 2 64-bit CPU and quad core ARM A57 complex CPU, and has 8 GB 128-bit LPDDR4 memory shared between CPU and GPU. At 170 mm $\times$ 170 mm $\times$ 50 mm, the dimensions of

the TX2 developer kit is considerably larger than the Intel NCS, but does not have the requirement of a host machine.

NVIDIA GTX 1080 (Reference)

As mentioned earlier, GPUs have been the driving force in the deep learning revolution. They are especially good at performing a large number of simple parallel computations efficiently in comparison to CPUs. Previously, GPUs were primarily used for graphics in computer games, but as the hardware became cheaper and more powerful, they were made viable to train and run DNNs. The aforementioned tasks have a number of similarities in parallel computations – computer games must transform a large number polygons in 3D, whereas CNNs run convolutions across an entire image, both through matrix computations. In this work, we evaluate the NVIDIA GTX 1080 containing 2560 CUDA cores, which is a popular GPU for deep learning practitioners.

3.3 Evaluation Overview

As mentioned in Section 3.1, all models take basis in pretrained models available from the TensorFlow [44] model zoo[1]. The so called *frozen* models are available in *pb* format that can be used directly to benchmark the GTX 1080 GPU and i7-7700K CPU. However, to evaluate the models on the TX2, NCS and NCS 2, they need to be appropriately converted using platform specific tools.

In case of the TX2, models run in three settings; (1) in the standard TensorFlow format, (2) by maximising the clock speed on the TX2, (3) and by optimising the models with the TensorRT (TF-TRT) package [55], which transforms and optimises the models, for example by fusing layers, such as Convolution and ReLU. Additionally, the precision of the model is changed from FP32 point to FP16, with minimal loss in accuracy. To run model inference on the NCS and NCS2, the models are converted to an Intermediate Representation (IR), consisting of an *xml* file to describe the model topology and a *bin* file containing model weights and biases. As mentioned in Section 3.2, this is accomplished using the OpenVINO toolkit [54], developed by Intel. Similarly to TF-TRT, this is accomplished by fusion of certain layers of the network, such as Convolution and BatchNormlisation or removing layers that are not used at test time, for example, the dropout layer. Likewise, the precision of the model is changed to FP16 in order to speed up inference and make the model compatible.

Evaluations are performed using TensorFlow 1.10.1 for most platforms. Additionally for the NCS and NCS2, OpenVINO 2018_R5 is used to optimize

[1] Available from: `https://github.com/tensorflow/models/\tree/master/research`

and run evaluation. However, TensorFlow 1.8 is used in case of TX2 as this was compatible with TensorRT 4.0.1, which is required to optimize models to TRT. To accelerate performance on TX2 and GTX 1080, we use CUDA 9.0 with CUDNN 7.0. The GTX 1080 and i7-7700k are evaluated on a machine containing 64GBs of RAM, running Ubuntu 16.04, while NCS and NCS2 are evaluated on a machine consisting of an i7-6700HQ CPU @ 2.60GHz and 16GBs of RAM. In all cases, evaluations are executed in Python 3.5.2.

The evaluations are run on images from the ImageNet dataset [2]. N images are loaded, where N is the batch size, and resized accordingly to the input size of the model. For NCS and NCS2, the batch size corresponds to the number of sticks that are run in parallel, asynchronously. We run inference for 100 iterations and calculate the mean inference time per image based on the total inference time and batch size. We evaluate inference time using batch sizes {1, 2, 3, 4, 8, 16, 32, 64, 128}, in case of NCS and NCS2, we evaluate inference time using 1, 2, 3 and 4 sticks in parallel. The entire evaluation procedure is summarised in Algorithm 1.

Algorithm 1: Evaluation procedure

Input: $model\_name,\ batch\_size,\ platform,\ imagepath$;
Output: $mean\_inference\_time$;
$model \leftarrow load(model\_name)$;
if $platform == tx2\ trt\ ||\ platform == NCS$ **then**
    $model \leftarrow convert\_model(model)$;
$images \leftarrow read\_images(batch\_size, imagepath)$;
$i \leftarrow 0$;
$total\_time \leftarrow 0$;
while $i < 100$ **do**
    $start\_time \leftarrow time()$;
    $run\_inference(model, images)$;
    $inference\_time \leftarrow \frac{time() - start\_time}{batch\_size}$;
    $i \leftarrow i + 1$;
    $total\_time \leftarrow total\_time + inference\_time$;
$mean\_inference\_time \leftarrow \frac{total\_time}{100}$;

4 Experimental Results

We perform experiments to conclude on the optimal model/platform selection within each task. Extensive plots are provided to aid selection based on platform price, inference time, and batch size. First, we plot the inference

time depended on batch size to conclude on platforms where an increase of batch size results in large decrease in inference time. Next, we compare inference time of different platforms across models of different complexities. Finally, we plot an FPS cost based on retail price and inference timings, to provide a more extensive overview and further aid decision making based on available resources and timing requirements. The FPS cost is calculated as the retail price divided by the number of FPS for at given platform/model combination. Since some of the plots show large numerical differences between platforms, whenever it makes sense, we plot values on a logarithmic scale. This is mostly the case, when comparing price and inference time.

As mentioned in Section 3.2, we run the TX2 in three settings; using the TF model with and without maximised clock and by converting to a TRT model. Furthermore, we run the i7 in three settings; using one, two and four cores. For an overview, all combinations of platforms and settings are summarised in Table G.5.

Platform	Setting
GTX 1080	TF model
TX2	TF model
TX2 MAX	TF model+Maximised clock
TX2 TRT	TRT model
NCS	IR model
NCS2	IR model
i7-1	TF model+single core
i7-2	TF model+dual core
i7-4	TF model+quad core

Table G.5: Overview of platforms and settings that are evaluated.

4.1 Classification

Figures G.1-G.3 show the inference timings based on batch size for each classification network. In Figure G.1, at batch size one, the inference time of TX2 MAX is comparable to that of i7-4 at 15 ms, while inference timings of TX2 TRT and NCS2 are comparable to that of i7-2 at 22 ms. Slower timings appear for the NCS which is more comparable to i7-1 at 37-41 ms. Compared to the reference GTX 1080, TX2 MAX and i7-4 are 13 ms slower while TX2 TRT, NCS2 and i7-2 are 20 ms slower. However, the NCS and NCS show considerable speed increases as batch size increases, resulting in the lowest inference timings at batch sizes three and four. At almost identical inference timings at batch size four, the NCS and NCS2 are the fastest edge platforms,

only slightly slower than the GTX 1080. Nonetheless, at batch size two, the NCS2 is considerable faster than the NCS. Meanwhile, TX2, TX2 MAX and TX2 TRT show less decreases in inference time, resulting in almost identical timings between the two former at batch size 128. Finally, at batch size two and onward, TX2 TRT is faster than TX2 MAX.

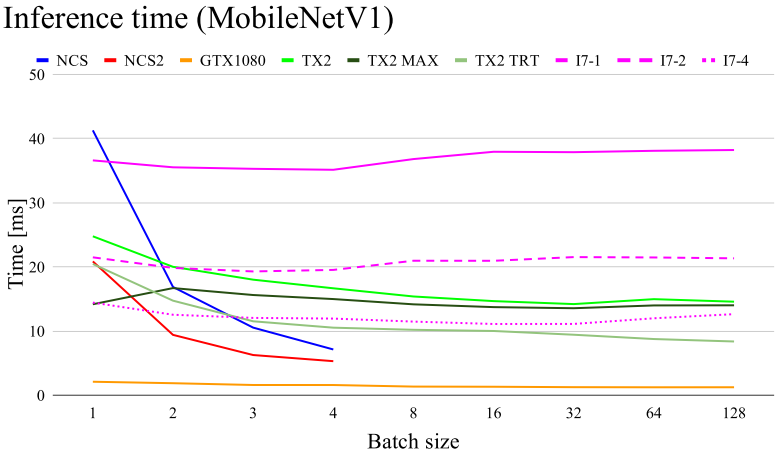


Fig. G.1: Inference timings for MobileNetV1 based on batch size.

Similar tendencies as that of MobileNetV1 in Figure G.1 is seen in Figure G.2. However, different from the MobileNetV1, NCS2 and TX2 TRT show almost identical timings at batch sizes 2-4, while they are almost equivalent to the GTX 1080 at batch size four. Meanwhile, TX2 MAX and TX2 show timings that are comparable to that of i7-4 until batch size four, from here, the two TX2 settings are faster than i7-4 with increasing margin up to batch size 128. Again, the slowest edge platform across at batch size one is the NCS, however, at batch size four the timings are comparable to those of TX2, TX2 MAX and i7-4. Interestingly, comparing NCS and NCS2, the former shows a much larger speed-up, as we increase batch size.

Timings for InceptionResNetV2 are shown in Figure G.3. For this network, timings of TX2 MAX and TX2 TRT were not possible due to memory limitations. Furthermore, timings of TX2 were only possible at batch size one. At batch size one, inference time of TX2 is comparable to NCS2 and i7-4, while NCS is almost four times as slow. Again, the speed-up of NCS is higher than that of NCS2 making the difference in inference time between the two sticks much lower at batch size four. However, the NCS2 is still considerable faster than the both the NCS and i7-4, and only slightly slower than the GTX 1080. Similarly, the NCS and NCS2 show speed increases as in Figures G.1 and G.2. At batch size one, inference time of the NCS is relatively high,

but has a significant improvement as the the batch size, i.e. number of sticks, increases.

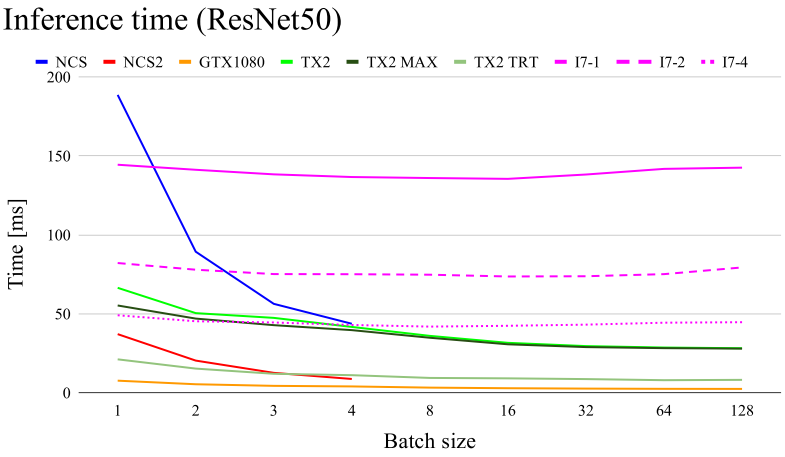


Fig. G.2: Inference timings for ResNet50 based on batch size.

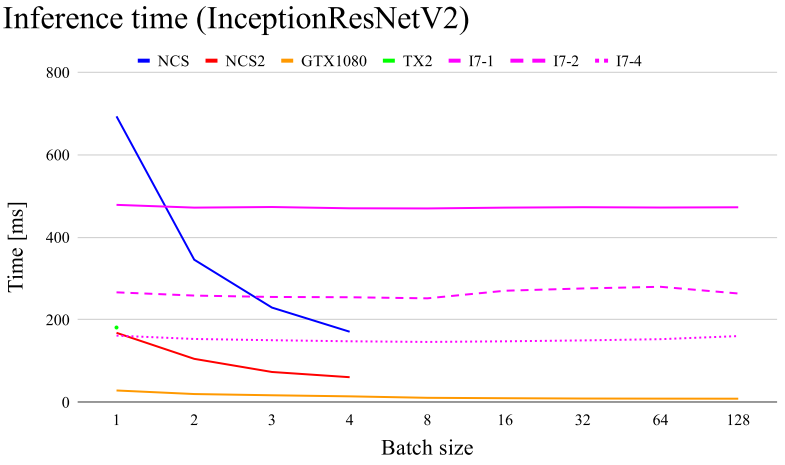


Fig. G.3: Inference timings for InceptionResNetV2 based on batch size.

Figure G.4 shows inference timings for each model in case of batch sizes one (left) and four (right). Naturally, inference time increases as model complexity increases, which is present in both figures. However, if we compare platforms across models, at batch size one, we see that the inference time of NCS2 on ResNet50 is comparable to those of i7-1 and NCS on MobileNetV1.

A similar tendency is between NCS and TX2 on InceptionResnetV2 and i7-1 and NCS on ResNet50. Additionally, at batch size four, both NCS2 and TX2 TRT on ResNet50 show inference timings lower than TX2 and TX2 MAX on MobileNetV1. Furthermore, the NCS2 on ResNet50 has almost identical inference time compared to NCS on MobileNetV1, while being faster than TX2 TRT and all i7 on MobileNetV1.

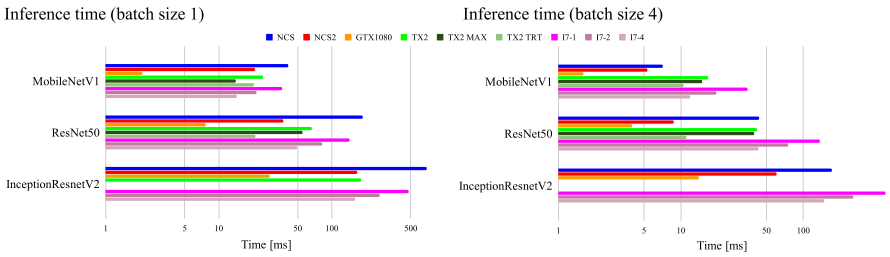


Fig. G.4: Comparisons of inference timings between classification models for batch sizes 1 (left) and 4 (right).

Figure G.5 shows the FPS cost of all three classification models. For each model, the left plot compares FPS cost to batch size, while the right plot compares FPS cost to FPS for all batch sizes.

For MobileNetV1, we see that the NCS and NCS2 are the most cost friendly edge platforms for batch sizes 1-4, yet at a somewhat consistent costs, not much higher than GTX 1080. At batch size four, the FPS costs of the two sticks are almost identical. While the FPS costs of the three CPU settings stay almost constant across batches sizes with, naturally, the higher number of cores resulting in lower costs, the FPS costs of TX2 and TX2 TRT decrease as batch size increases. At batch size one, the FPS costs of TX2 and TX2 TRT are higher than both i7-1 and TX2 MAX, but as batch size increases, the FPS cost of TX2 TRT becomes almost identical to that of i7-4 and almost twice as low as that of TX2 MAX. This indicates that optimising the models with TensorRT, combined with batching, significantly decreases the price of running a MobileNetV1 model on the TX2 platform.

Comparing FPS cost simultaneously to FPS and batch size, we see that most platforms follow linear decreasing tendencies between FPS cost and FPS as we increase batch size. At batch size one, the FPS cost of NCS2 is half that of i7-2 at same FPS. Similarly, at batch size two, the FPS cost of NCS2 is more than half that of TX2 TRT at almost identical FPS, while at batch sizes three and four, NCS2 achieves higher FPS, while maintaining FPS cost. Furthermore, FPS and FPS cost of NCS2 at batch size three is almost identical to NCS at batch size four, while the FPS cost of NCS2 at batch size one is slightly slower than that of NCS, while FPS is considerably larger. While multiple NCS or NCS2 do not result in a lower FPS cost, large increase

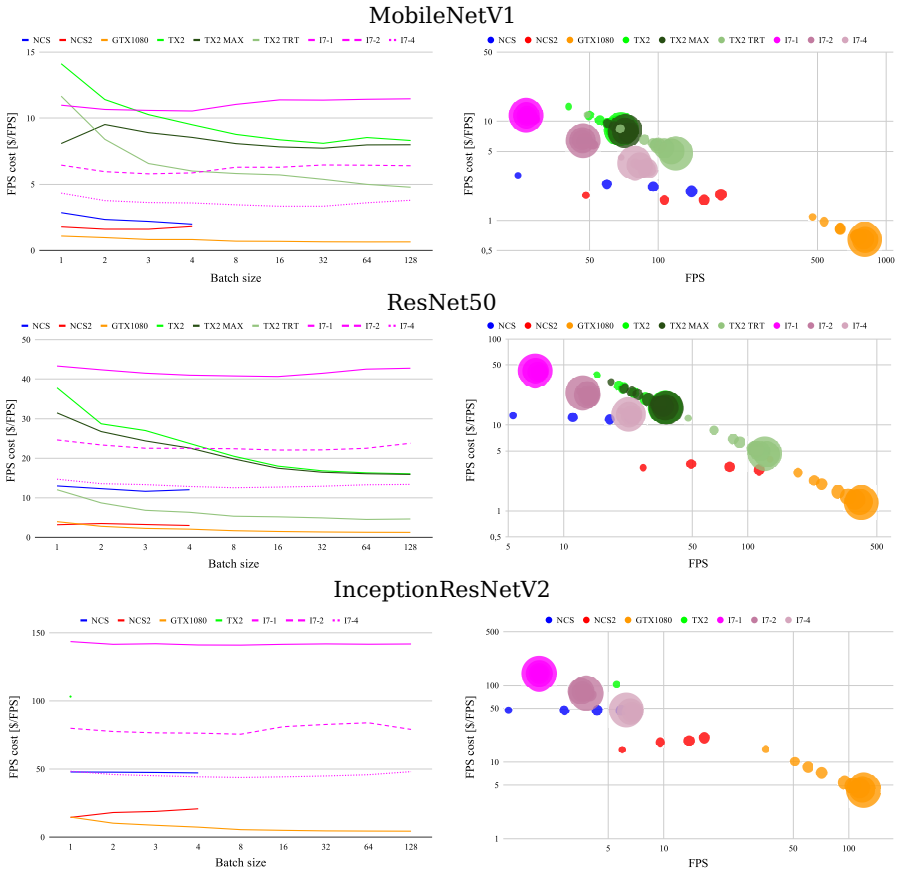


Fig. G.5: FPS cost of classification models based on batch size and FPS.

in FPS is achieved. Finally, at batch size four, the FPS cost of TX2 TRT is identical to those of i7-2, TX2 and TX2 MAX at larger batch sizes, while TX2 TRT at larger batch sizes is faster than i7-4 with an almost identical FPS cost.

In case of ResNet50, we see that the FPS cost of NCS2 at batch size one is even lower than that of GTX 1080. As batch size increases, the FPS cost of GTX 1080 drops below that of NCS2, however, the two platforms have almost identical values at batch size four. For this network, the differences in FPS cost between NCS and NCS2 are much higher, with the FPS cost of the former being identical to i7-4, however, still lower than TX2 and TX2 MAX. We also see that the FPS cost of TX2 TRT is much lower than TX2 and TX2 MAX at batch size one, while it is only slightly higher than NCS2 and GTX 1080 at batch sizes 8-128. Compared to MobileNetV1, we see for ResNet50 a significant decrease in FPS cost occurs when optimising models with TensorRT on the TX2 platform. Despite the fewer FPS of TX2 TRT, the

largest decreases in FPS costs are seen in case of TX2 and TX2 MAX. At batch sizes 1-4, the FPS costs are either higher than or on par with i7-2, while at larger batch sizes, the costs are almost identical to those of i7-4 and NCS.

Considering also FPS, we see that, at larger batch sizes, TX2 and TX2 MAX are faster compared to i7-4 and NCS, while maintaining FPS costs. We also see that at a slightly lower FPS cost, the NCS2 at batch size four is as fast as TX2 TRT at larger batch sizes, while having an FPS cost almost identical to GTX 1080 running batch size one. Furthermore, at batch sizes 2-4, not only is the FPS cost of NCS2 much lower, FPS is also higher compared to all i7, TX2, TX2 MAX and NCS.

Finally, for InceptionResNetV2, the GTX 1080 and NCS2 have similar FPS costs at batch size one. However, while the FPS cost of NCS2 increases slightly at higher batch sizes, the FPS cost of GTX 1080 is decreasing. Nonetheless, compared to remaining edge platforms, the FPS cost of NCS2 is much lower. We also see that the FPS cost of NCS is slightly higher than that of the i7-4, which is opposite to the situations on ResNet50 and MobileNetV1. While the TX2 could only run at batch size one, the FPS is higher than that of i7-2.

Looking at the inference price of InceptionResNetV2, there is a clearer distinction between the GTX 1080 and NCS2 at batch size one. While prices are equal, the GTX 1080 is considerably faster per image, as shown in Figure G.3.

Figure G.6 shows both the FPS cost (top) and retail price (bottom) as a function of FPS for all platform/model combinations. Plots are shown for batch sizes one (left) and four (right). With these figures, if the budget is known for a deep learning system, it is possible to infer how complex a model can be run and at what speed. For example, at batch size one we can see that if there is a budget of roughly $100, an NCS can be purchased, which allows a system to perform classification at almost 50 FPS using MobileNetV1. The FPS cost of this platform/model combination can then be inferred on the other plot, in this case, an NCS2 running MobileNetV1 seems to be a good choice with an FPS cost only slightly higher compared to the GTX 1080. Differently, if we wish to run a more complex model at same FPS, this can be done using TX2 TRT, however, at a higher FPS cost. At same FPS costs, i7-4 at fewer FPS is also an option, comparing the retail price to that of the TX2 TRT, it is much lower, thus, it comes down to requirements on speed. Finally, if high precision is a requirement, the optimal options are the NCS, NCS2 or the i7. While FPS cost of both NCS and NCS2 are lower than those of i7, FPS cost and FPS of NCS2 are much lower and higher, respectively, compared to NCS, although, at a slightly higher retail price.

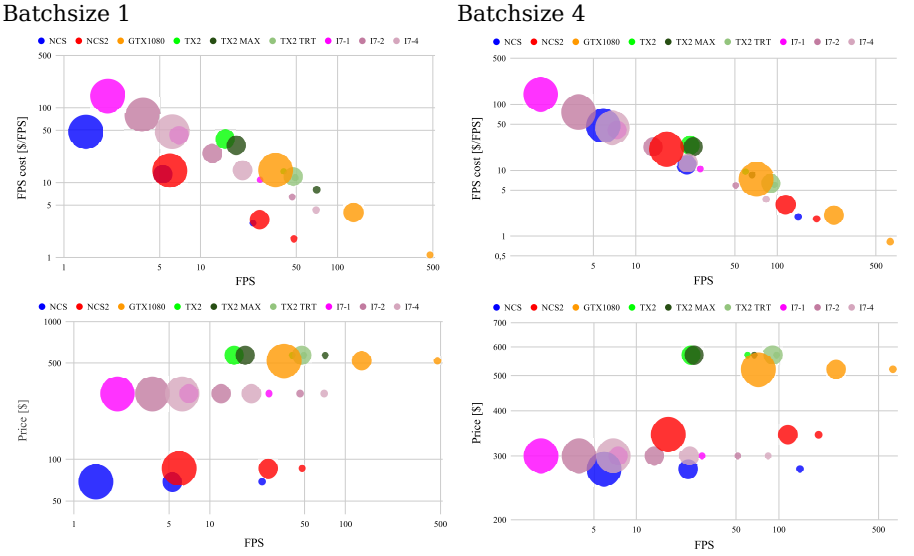


Fig. G.6: Comparison of FPS cost (top) and retail price (bottom) based on FPS, for batch sizes one and four. Small bubbles indicate MobileNetV1, middle-size bubbles indicate ResNet50, and large bubbles indicate InceptionResNetV2.

4.2 Object Detection

Figures G.7 and G.8 show inference timings, dependent on batch size, on SSD with backbones MobileNetV1 and InceptionV2, respectively. Similarly to the classification models, combining multiple NCS's or NCS2's results in the largest decreases, with inference times below 25 ms. At batch size four, inference time of NCS2 is almost on par with the GTX 1080. In case of batch size one, the NCS2 is twice as fast as NCS, however, with the combination of multiple sticks, this difference is drastically decreased. Meanwhile, the i7 only shows minor decreases in inference time, independent of number of cores utilised, while inference timings across all batch sizes are decreased as the number of utilised cores increases. While i7-1 clearly is slowest at batch sizes larger than one, i7-4 is faster than all settings of the TX2 in case of both SSD MobileNetV1 and SSD InceptionV2. Additionally, while being faster than both NCS and NCS2 at batch size one, it is slower than the NCS2 if the batch size is larger than two. Interestingly, when using a TX2, increasing batch size does not necessarily results in faster inferences. Using MobileNetV1 as backbone, both TX2 and TX2 TRT show either similar or decreased timings up to a batch size of three, however, increasing the batch size further results in increased inference timings. Only TX2 MAX shows constant decreases in inference timings by increasing batch size. Similar tendencies are shown in Figure G.8, however, to a minor extent.

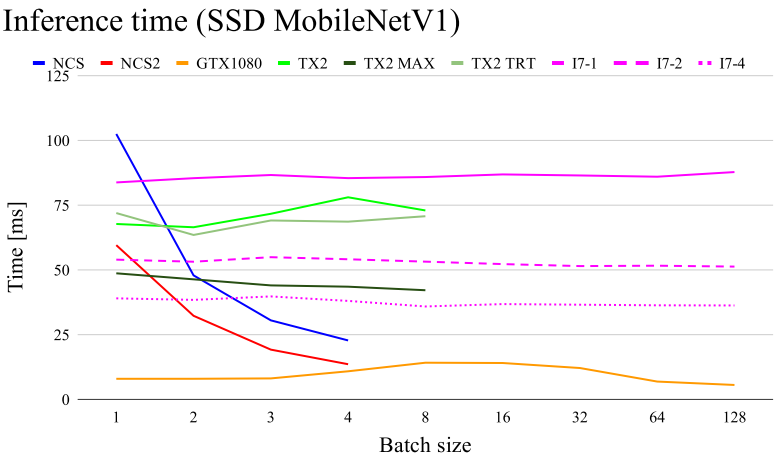


Fig. G.7: Inference timings for SSD MobileNetV1 based on batch size.

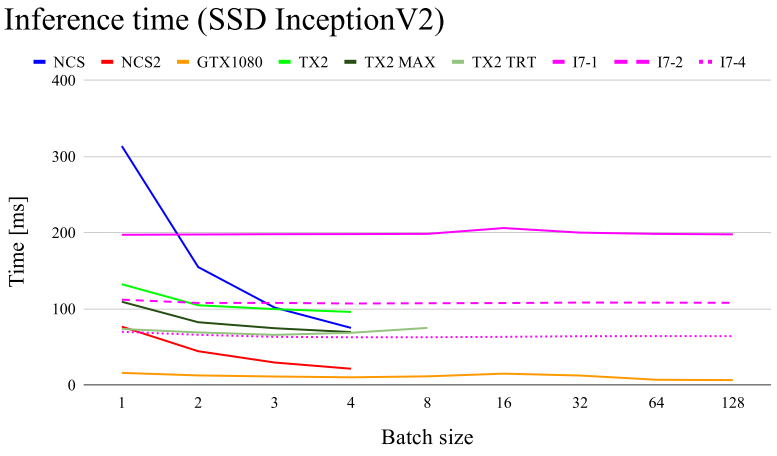


Fig. G.8: Inference timings for SSD InceptionV2 based on batch size.

As seen in both Figure G.7 and G.8, the NCS and NCS2 are limited to a batch size of four, while the batch size of TX2 is also limited to eight and four, respectively. While the former is limited by number of available sticks, the latter is limited by the available memory on the board, and only using a larger GPU, such as the GTX 1080, or a CPU with enough available RAM, allows for larger batch sizes.

Figure G.9 provides an overview inference timings across platforms and detection models in case of batch sizes one and four. While most platforms

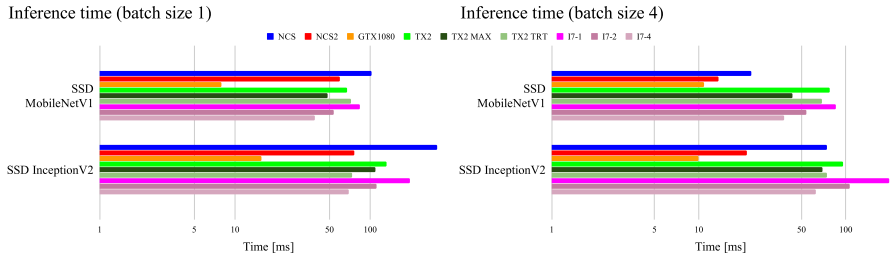


Fig. G.9: Comparisons of inference timings between SSD MobileNetV1 and SSD InceptionV2 for batch sizes one and four.

show increasing inference timings as the complexity of the detection model increases, TX2 TRT shows almost constant inference timings between SSD MobileNetV1 and SSD InceptionV2. The reason is probably a more efficient model conversion in case of the latter. This provides TX2 TRT with an advantage compared to TX2 MAX, where TX2 MAX goes from being faster on SSD MobileNetV1 to being slower on SSD InceptionV2. It is also worth noting a smaller increase in inference time of NCS2 compared to the NCS. This results in inference timings of the NCS2 using InceptionV2 as backbone that are lower than those of the NCS using MobileNetV1 as backbone, independent of batch size. Finally, the NCS2 on SSD InceptionV2 at batch size one is also faster than i7-1 on SSD MobileNetV1, while at batch size four it is faster than all but the GTX 1080.

Figure G.10 shows FPS cost, depending on both batch size and FPS, for each platform and detection model. For SSD MobileNetV1, the FPS costs of NCS and NCS2 drop steadily with the cost of the latter ending up lower than that of GTX 1080 at batch size four, while the cost is almost identical between the NCS and NCS2. Meanwhile, FPS costs of TX2 and TX2 TRT are increasing as batch size increases, while FPS cost of TX2 MAX steadily decreases with increasing batch sizes. At batch size eight, the FPS cost of TX2 MAX is slightly lower than that of i7-1, however, still more than twice as high as that of i7-4. On SSD InceptionV2, on the other hand, FPS cost of TX2 TRT increases at batch sizes larger than three, ending up at an FPS cost identical to that of TX2 MAX at batch size four. This is also due to much larger decreases in FPS costs for TX2 and TX2 MAX compared to SSD MobileNetV1. However, FPS costs of all TX2 settings are still higher than those of i7-2 and i7-4. Similarly to SSD MobileNetV1, the FPS cost of NCS2 is almost identical to that of GTX 1080, meanwhile for this network, difference in FPS cost between NCS and NCS2 are much larger, with former being on par with that of i7-4.

Additionally comparing FPS cost and FPS, interestingly in case of SSD MobileNetV1, both FPS cost and FPS of the NCS2 are comparable to those of GTX 1080 at batch sizes between four and 32, while the FPS cost of NCS is

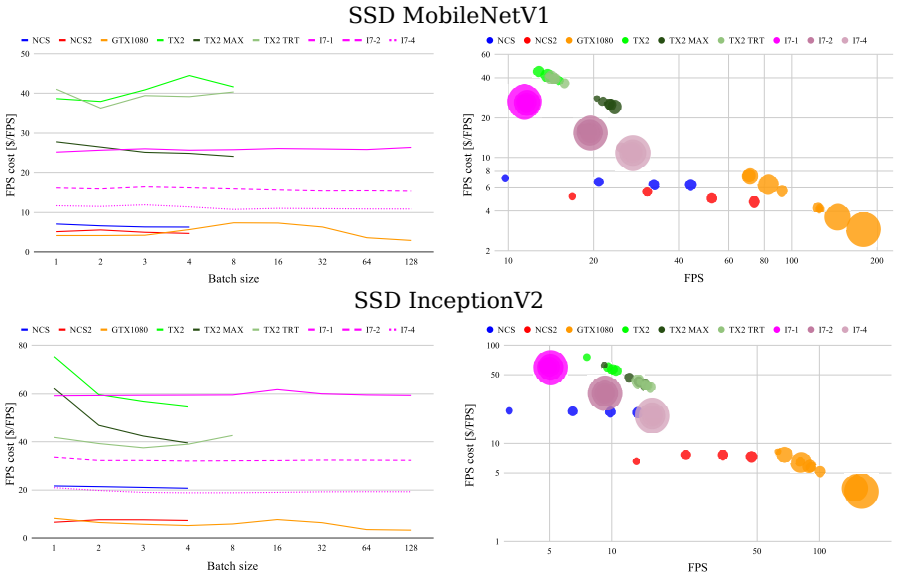


Fig. G.10: FPS cost of SSD MobileNetV1 and SSD InceptionV2 based on batch size and FPS.

similarly low, but at fewer FPS. Moreover, FPS cost and FPS of NCS2 at batch size two is similar to those of NCS at batch size three, while FPS cost and FPS of NCS2 at batch sizes three and four both are lower than those of NCS at batch size three. Like Figure G.5, linear correlations between FPS cost and FPS are shown across all but i7, where FPS cost is constant for all batch sizes. However, while i7-1 is slightly slower at a lower FPS cost compared to TX2 and TX2 TRT, FPS costs of i7-2 and i7-4 are lower than that of TX2 MAX at almost identical FPS.

In case of SSD InceptionV2, a larger gap is shown between NCS and NCS2 in terms of both FPS cost and FPS, with the latter having a much lower FPS cost compared to the remaining edge platforms. Furthermore, FPS costs of the NCS2 are similar to those of GTX 1080 at batch sizes less than 64. At batch size one, FPS of NCS2 is almost similar to those of NCS, TX2 TRT and TX2 MAX running batch size four, while the FPS cost of NCS2 is much lower. Meanwhile, both FPS cost and FPS of TX2 MAX and TX2 TRT are almost similar across batch sizes.

To summarise FPS cost of platforms across detection models, Figure G.11, shows the FPS cost and FPS for all platform/model combinations (top) in case of batch sizes one (left) and four (right), while also directly comparing retail price and FPS (bottom). At batch size one, the FPS cost of NCS2 on SSD InceptionV2 is similar to that of GTX 1080, however, at much fewer FPS. Furthermore, the FPS cost of NCS2 is similar to that of NCS on SSD

MobileNetV1, while FPS is higher, additionally, the FPS cost is much lower than those of i7-1, TX2 and TX2 TRT on SSD MobileNetV1, while FPS is almost similar. Connecting these observations to actual retail price, NCS2 is a much cheaper choice compared to TX2. FPS cost and FPS of TX2 TRT on SSD InceptionV2 is similar those of TX2 and TX2 TRT on SSD MobileNetV1, therefore, it makes sense to consider a more complex model if it is run on the TX2 using a converted TRT model. However, at same FPS, SSD InceptionV2 on i7-4 is similar to TX2 TRT, but at a lower FPS cost.

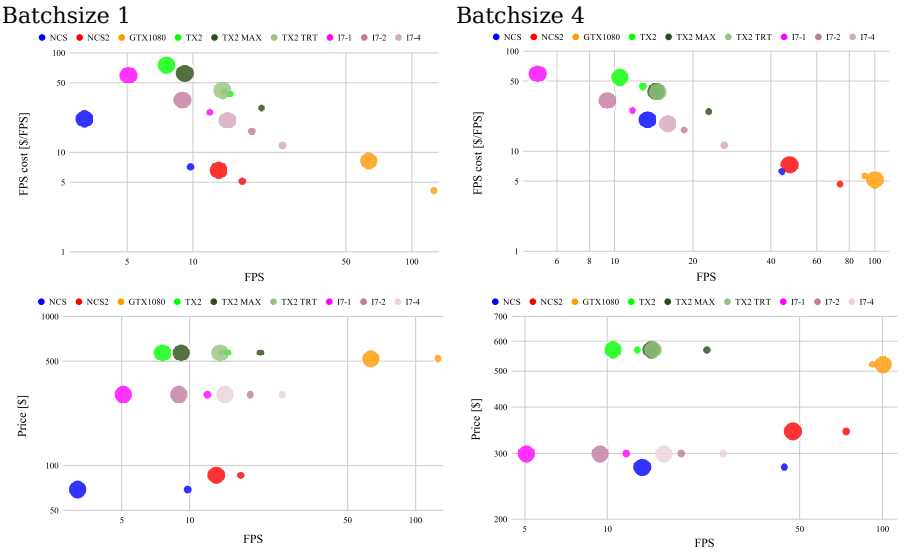


Fig. G.11: Comparison of FPS cost (top) and retail price (bottom) based on FPS, for batch sizes one and four. Small bubbles indicate SSD MobileNetV1 and middle-size bubbles indicate SSD InceptionV2.

Finally, at batch size of four, FPS of most platforms running SSD InceptionV2 lies within a smaller range of 10-20 ms. Again, the FPS cost of NCS2 on SSD InceptionV2 is identical to that of NCS on SSD MobileNetV1, while being slightly faster. Furthermore, on SSD MobileNetV1, the FPS cost of NCS2 is lower than that of GTX 1080, at slightly fewer FPS. Different from batch size one, TX2 TRT and TX2 MAX show similar FPS costs and FPS, both favourable to TX2. Additionally, the FPS cost of NCS on SSD InceptionV2 is lower than TX2 on SSD MobileNetV1, while showing similar FPS. Comparing to retail price, when combining multiple sticks, the NCS and NCS2 do not have the same advantage price compared to the i7, since a larger batch size requires multiple sticks. Nonetheless, the combination of four sticks is still much cheaper than the TX2, and results in more FPS. Finally, running SSD InceptionV2 at a batch size of four on the i7, using either two or four cores, is more favourable compared to the TX2 running SSD MobileNetV1.

4.3 Semantic Segmentation

Since it was only possible to run segmentation models with a batch size of one, we have summarised inference timings of both DeepLabV3 models in Figure G.12. Contrary to both the classification and detection models, NCS and NCS2 perform worse compared to other platforms. Additionally, while NCS2 shows lower inference timings on DeepLabV3 Xception65 compared to NCS, the situation is reversed in case of DeepLabV3 MobileNetV2. Meanwhile, nothing is achieved by maximising the clock on the TX2 when comparing TX2 and TX2 MAX, while the both are slightly faster than the i7. Finally, increasing the number of utilised cores on the i7 does not affect inference time on either of the models.

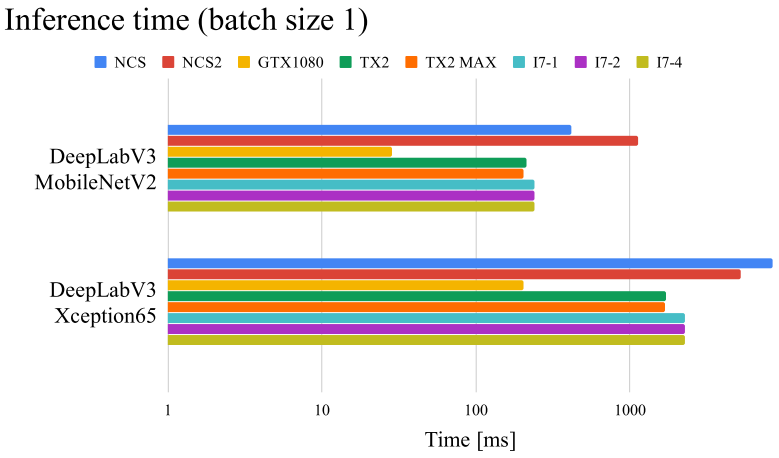


Fig. G.12: Inference timings of DeepLabV3 with backbones MobileNetV2 and Xception65 using a batch size of one.

Figure G.13 compares FPS cost and FPS between platforms for the two segmentation models at batch size one. On DeepLabV3 MobileNetV2, FPS of TX2 and TX2 MAX are slightly higher than those of NCS and NCS2, while the FPS costs are much higher compared to the NCS. Different from other tasks, the FPS cost of NCS is much lower than that of NCS2, while also being faster. In between, the FPS cost of i7 is lower than those of TX2 and TX2 MAX, while having almost similar FPS. On DeepLabV3 Xception65, an almost identical pattern is seen, although, having a lower FPS cost in case of NCS2 compared to NCS. While the FPS cost of the sticks are smaller than those of the TX2, TX2 MAX and i7, they are much slower. For either models, no platform exceeds five FPS, indicating the need of a much more powerful platform in case of real time requirements on speed.

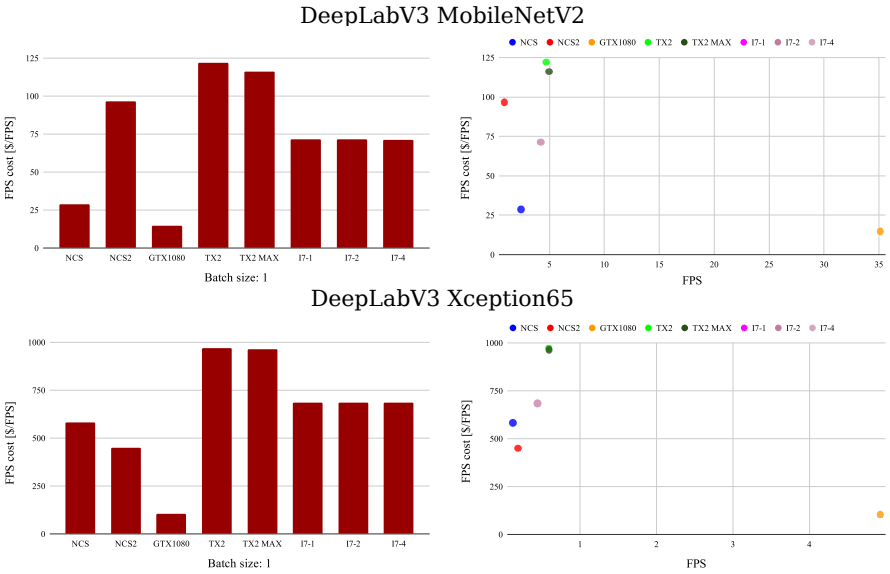


Fig. G.13: FPS cost based of DeepLabV3 models based on batch size and FPS at batch size one.

Figure G.14 shows the FPS cost compared to FPS between platforms and the two segmentation models. Not surprisingly, large differences in FPS costs are seen between the two models. While the FPS of TX2 and TX2 MAX on DeepLabV3 Xception65 are only slightly higher than that of NCS2 on DeepLabV3 MobileNetV2, the FPS cost of the NCS2 is much lower, due to the much higher retail price of the TX2.

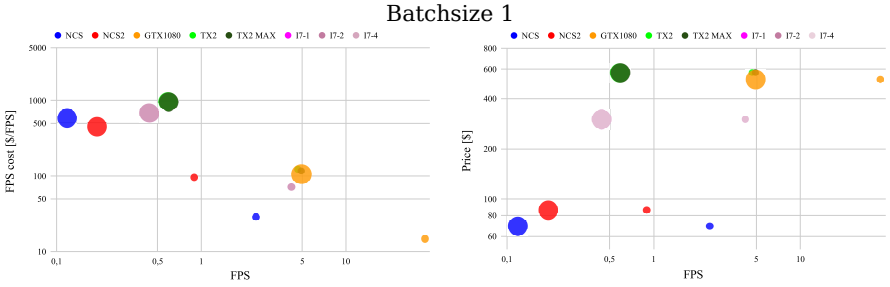


Fig. G.14: Comparison of FPS cost (left) and retail price (right) based on FPS at batch size one. Small bubbles indicate DeepLabV3 MobileNetV2 and middle-size bubbles indicate DeepLabV3 Xception65.

Overall, semantic segmentation models are not yet suited for edge platforms if real time speed is a requirement. If not, apart from the GTX 1080, either the NCS or NCS2 provide the lowest FPS cost, depending on which model to deploy. If budget is not a requirement, the i7 or TX2 is the optimal

choice.

4.4 Comparison of Tasks

We compare results from Figures G.6, G.11 and G.14 to conclude which platforms are more suited for specific tasks. If multiple NCS2 are combined, the platform is favourable in terms of both speed and price, to run classification or detection, independent of model complexity. Having a single NCS2, FPS performance on detection is still comparable to running TX2 TRT at batch size one, however, on classification TX2 TRT outperforms NCS2 in FPS. For both tasks, however, the FPS cost of NCS2 is still much lower. Nonetheless, on both classification and detection, TX2 TRT compares favourable to TX2 and TX2 MAX. Finally, segmentation is more suitable for the i7 or TX2, however, at a higher price compared to NCS2.

4.5 Inference Analysis

To dive deeper into the differences amongst the platforms we profile each of the networks across all platforms to see where differences lie in operation allocation. We use the TensorFlow profiler for the GTX 1080, TX2, TX2 TRT and i7, whereas, for the NCS and NCS2 we use the Deep Learning Workbench in OpenVINO. For each instance we visualise the operations as the top five for each platform and combine the remaining timings into one which we denote as *Other*.

Firstly, in the case of MobileNetV1 we show the operation allocations in Figure G.15. The top-5 operations for the GTX 1080 and TX2 are largely similar with a significant portion of time being spent on *Conv2D*, *FusedBatchNorm* and *Relu6*. A slightly larger share of time is spent on *Conv2D* with respect to GTX 1080, whereas for the TX2, *FusedBatchNorm* accounts for twice as much of its overall allocation. For the i7 the allocations are slightly different, firstly, it can be seen that the *Conv2D* operation does not appear in the top-5 timings. Rather, *Relu6* accounts for roughly 50% of the timing and *DepthwiseConv2dNative* around 33%. For the remaining platforms the operations are not directly comparable. For TX2 TRT, several operations are shown as one through the *TRTEngineOp* optimised graph and here we see that almost 90% of the time is spent. Other operations only account for a few percentage of the time, such as *Mul* and *Softmax*. For the two NCS sticks, the large majority of time is spent on the *Convolution* operation, especially in the case of the NCS2 but less so for the NCS as *Relu6* accounts for almost 14%.

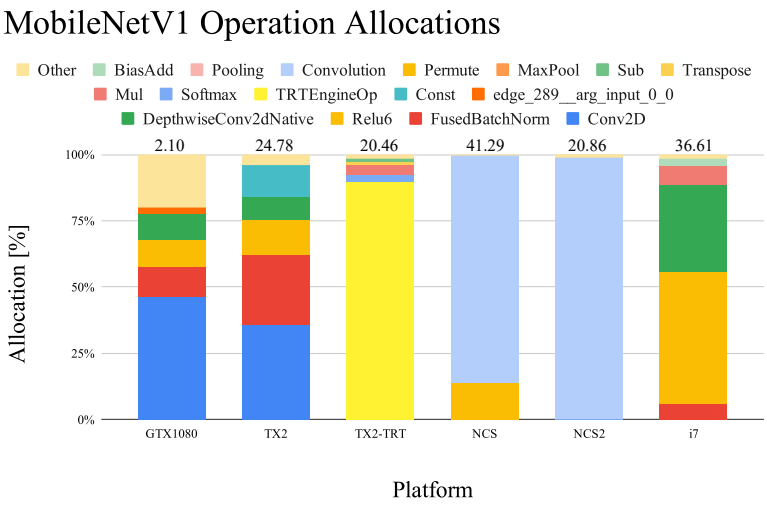


Fig. G.15: Operation allocation for MobileNetV1. Numbers above bars indicate total time in ms.

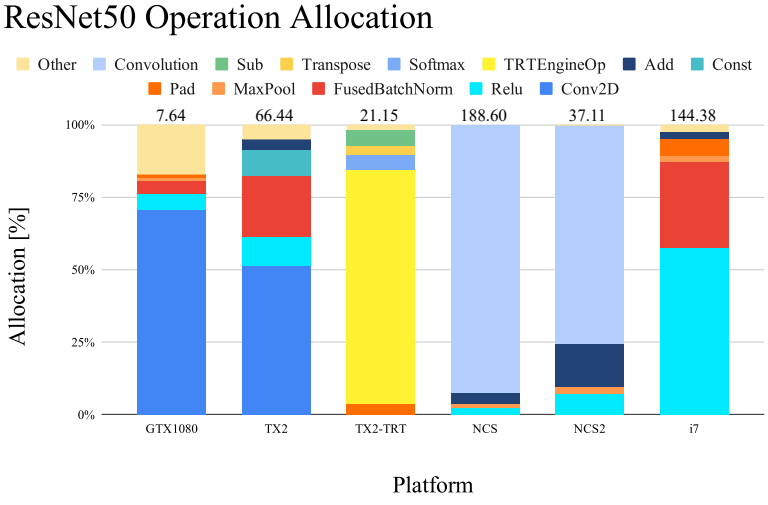


Fig. G.16: Operation allocation for ResNet50. Numbers above bars indicate total time in ms.

The timings for ResNet50 show similar trends in Figure G.16 as that seen for MobileNetV1. An even larger proportion of time is spent on the *Conv2D* operation for both GTX 1080 and TX2. For the TX2 TRT, the large majority time is also spent on *TRTEngineOp*. The NCS sticks again spend the majority

on the *Convolution* operation, however, in this case the NCS spends a larger portion compared to that seen between the two for MobileNetV1. Finally, for the i7, the *Relu* operation is where much of the computing is done.

The InceptionResNetV2 timings in Figure G.17 lose the TX2 and TX2 TRT due to insufficient memory. However, for most platforms, similar timings occur. GTX 1080 spends the majority of time on *Conv2D*, the NCS sticks on *Convolution* and the i7 again spends a lot of time on the *Relu* operation.

Overall for classification, the TX2 seems to allocate most time on *FusedBatchNorm* compared to other platforms, which combined with *Conv2D* and *Const* make up 75% of the total time. To reduce FPS cost on the classification models focus, thus, should be on optimising these operations. In case of TX2 TRT, most operations are already optimised, to reduce the FPS cost of platform, more simple operations should be optimised, such as *Softmax* and *Sub*. Finally, on NCS and NCS2, *Convolution*, *Add* and *Relu6* should be maximally optimised to increase FPS, thus, reduce FPS cost.

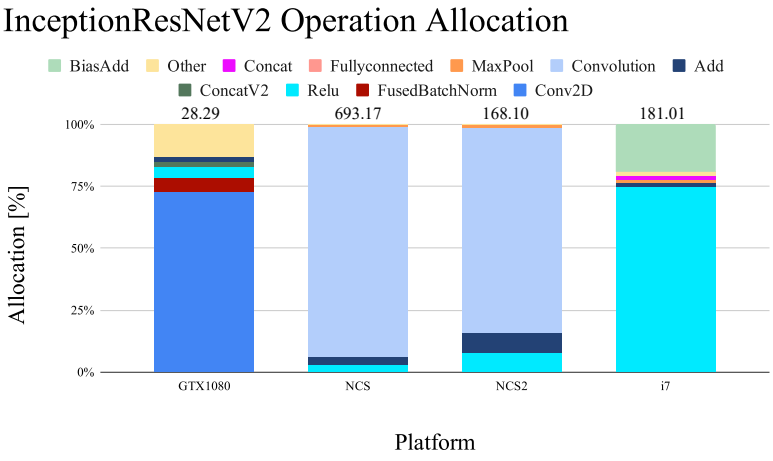


Fig. G.17: Operation allocation for InceptionResNetV2. Numbers above bars indicate total time in ms.

Figures G.18 and G.19 show the timings for the SSD networks. Here, we can see the differences in the more complex architectures present for object detection. In general, less time is spent on the convolution operations and more spread across multiple in comparison to the classification timings. Firstly, for SSD MobileNetV1 in Figure G.18, *Conv2D* accounts for 10-25% TX2 and GTX 1080 timings, respectively, with more emphasis being on other operations. In regards to the TX2 TRT, considerable less time is spent on the *TRTEngineOp* operation, indicating that less in the SSD networks can be optimised by TensorRT.

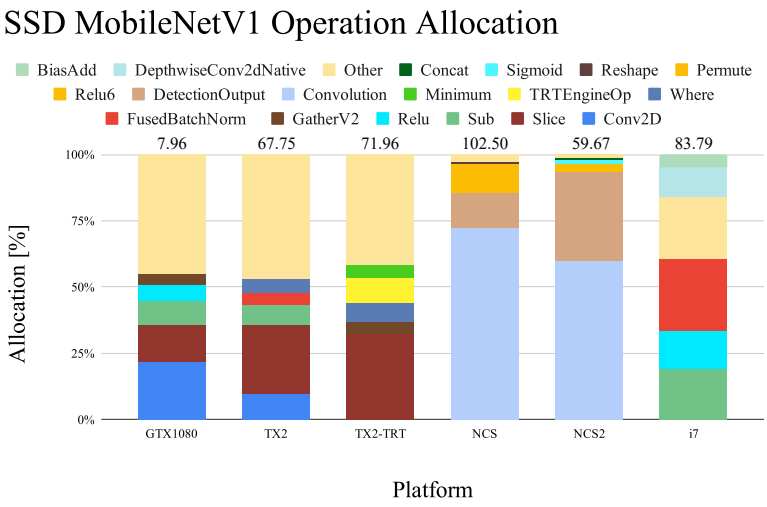


Fig. G.18: Operation allocation for SSD MobileNetV1. Numbers above bars indicate total time in ms.

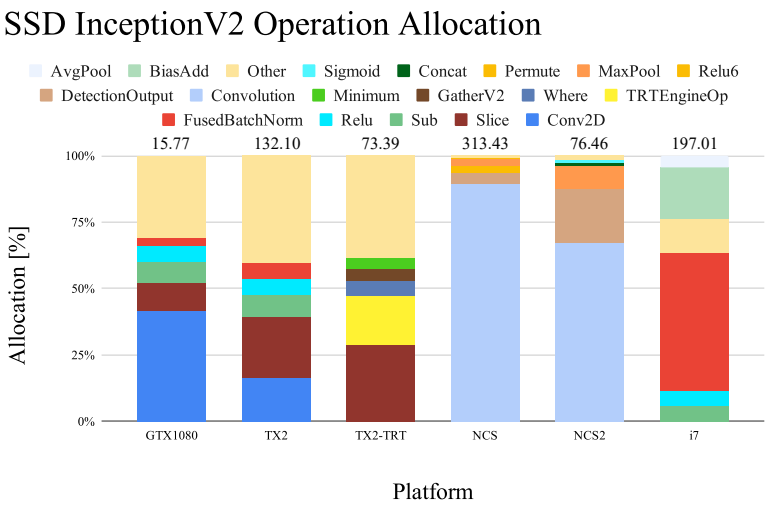


Fig. G.19: Operation allocation for SSD InceptionV2. Numbers above bars indicate total time in ms.

Instead, the largest operation is *Slice* followed by the grouping within *Other*. For both NCS and NCS2, the most time consuming operation is still

Convolution that, however, does decreases slightly as *DetectionOutput* is now introduced. Finally, again, the i7 is considerably different to the other platforms with timings being split between *FusedBatchNorm*, *Sub* and *Other* operations.

SSD InceptionResNetV2 in Figure G.19 is similar to the MobileNetV1 variant. Less time is spent on *Conv2D* for GTX 1080 and TX2, and also less on *TRTEngineOp* for the TX2 TRT. The NCS sticks are similar in that *Convolution* is still the largest operation, while the i7 spends around 50% of the time on *FusedBatchNorm*.

Different from classification models, the TX2 spends a much larger amount on non-convolution operations. Especially, *Slice* takes up a large amount of the total time, while several minor operations also should be optimised to reduce FPS cost, as indicated by *Other*. Compared to classification, several operations should be modified to be compatible with TensorRT. Similar to TX2, the *Slice* operation is the most critical, combined with several minor operations included in *Other*. While primarily the convolution operations make up the largest amount of computational time on the NCS and NCS2, different from classification, the postprocessing operations in *DetectionOutput* should be analyzed to identify ones that can be optimise to reduce FPS cost.

The final timings for the Deeplab variants are shown in Figures G.20 and G.21. Similar tendencies are seen as those of the detection networks in comparison to classification. For both DeepLabV3 MobileNetV2 and DeepLabV3 Xception65, less time is spent on *Conv2D*, however, in comparison to the SSD models, a larger proportion of time is spent on the operation for the TX2 board in comparison to the GTX 1080. A large portion of time is spent on the *Convolution* operation for the NCS and NCS2 on both DeepLab models. Finally, a more even spread occurs for the i7.

To make DeepLabV3 models more appealing to the TX2, similar to classification models, focus should be on optimizing the *Conv2D* and *FusedBatchNorm* operations. Likewise, *Convolution* and *Relu6* should be optimised for the NCS and NCS2.

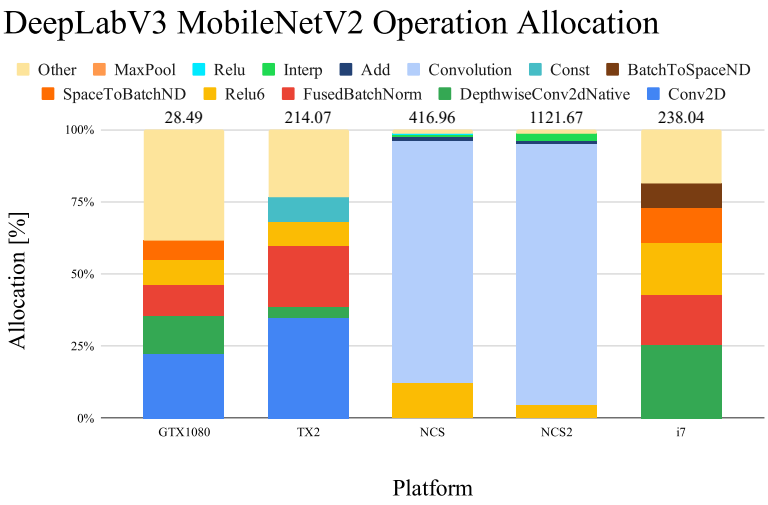


Fig. G.20: Operation allocation for DeepLabV3 MobileNetV2. Numbers above bars indicate total time in ms.

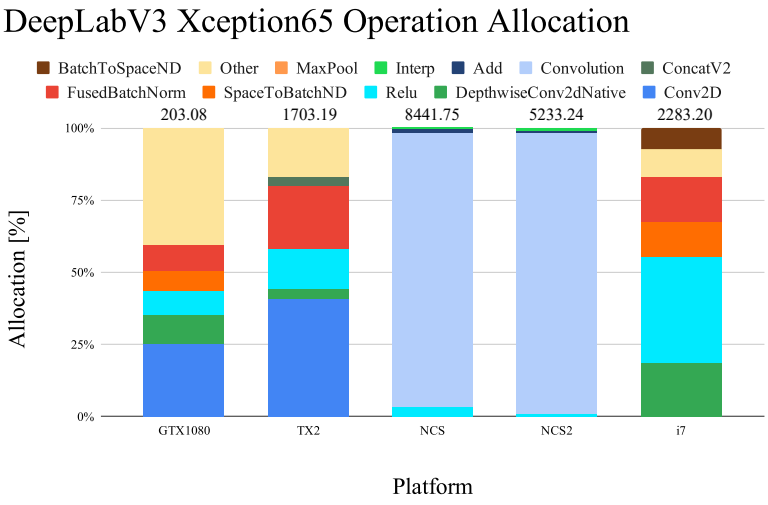


Fig. G.21: Operation allocation for DeepLabV3 Xception65. Numbers above bars indicate total time in ms.

5 Discussion

Even though, this work aims to aid faster decision making when purchasing a new platform, depending on timing and budget requirements, there are a few things that has not been taken into account, and should also be considered in the decision making process. As mentioned in Section 3.2, unless the development kit of the TX2 is purchased, all the platforms require a host machine to run. While this makes the platform evaluations more comparable, the price of a host machine differs between the platforms. In case of NCS or NCS2, a small computer, such as the Raspberry Pi will be enough to run inference, while the i7 requires a larger and more expensive composition of hardware. Finally, the TX2 module requires a carrier board to run.

While price and inference time are some of the most important factors when purchasing hardware, we have not included power consumption, which is also an important factor, especially, in case of edge platforms. To this extent, it would be interesting to compare power consumption based on batch size between models, to identify power requirements of running certain model on specific platforms and if how power consumption increases with batch size. For a more comprehensive overview of power consumption of different models, we can refer to [16]. However, they only ran models on a TX1 board.

As the focus has been on inference timing and FPS cost of various platforms for deep learning, the choice was to simply evaluate pretrained models from a popular framework, which in this case is TensorFlow. However, multiple other options exist, each with numerous options for pretrained models, such as Pytorch [56] or Caffe [43]. Each of these frameworks vary in their implementations, which naturally will affect potential timings. In this case, we chose the TensorFlow due to its popularity but also because of the extensive options of their pretrained models.

Finally, while we evaluate platforms as hardware that can be purchased, more companies offer cloud computing solutions to have large amounts of processing power without buying expensive hardware. It would be reasonable to include cloud solutions in comparison to other edge platforms in this work, however, both based on the period of time the hardware is running inference and how cloud solutions are paid, one platform quickly becomes in favour of another. While the price of some cloud solutions, for example Google Cloud, is based on an per-hour pricing [10], other solutions, such as IBM's Watson [57] is based on the number of inferences.

6 Conclusion

In this work, we have evaluated different edge platforms, including Jetson TX2, NCS and NCS2 and, finally, an i7 CPU, within object classification, object detection and semantic segmentation. We have analysed correlations between inference time and retail price as an FPS cost for models of different complexities withing the three tasks, to aid decision making of platform/model selection based on requirements. To that end, we also considered different batch sizes, to identify cases in which a larger batch size is favourable, when considering budget and timing requirements. Finally, we have analyzed allocation of DNN operations and compared this to the FPS cost. As a reference, all results of the edge platforms was compared with evaluations performed on a GTX 1080.

On classification, TX2 TRT is the optimal choice if a model runs at batch size one, and only speed is a requirement. However, if budget is limited, the NCS2 comes out as the better choice. Further, this is also the case for larger batch sizes, where the combination of multiple NCS2 is both cheaper and faster than compared to TX2, while being only slightly more expensive than the i7. For detection, a similar pattern is shown. However, at batch size one, differences between NCS2 and TX2 TRT in terms of FPS are much less, making the NCS2 favourable, independent of the number of sticks purchased. Finally, edge platforms are not yet suited for semantic segmentation, since only the GTX 1080 shows real-time inference timings. On the other hand, if real-time inference is not a requirement, either the NCS or NCS2 is the optimal choice in a strict budget, while TX2 is optimal in case of speed requirements.

Analyzing the allocation of DNN operation across platform/model combinations, we have shown that several operations in detection and segmentation models should be made compatible with TensorRT to increase FPS, thus, reduce the FPS cost, while primarily *Convolution* and *Relu* operations should be optimised for NCS and NCS2 to speed up inference.

References

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[4] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[5] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[6] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proc. ECCV*, 2018, pp. 116–131.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[8] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018.

[9] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proc. ICCV*, October 2019.

[10] Google, "Cloud machine learning engine," https://cloud.google.com/ml-engine/, 2019, accessed: 12 July 2019.

[11] Microsoft, "Microsoft azure," https://azure.microsoft.com/, 2019, accessed: 12 July 2019.

[12] Amazon, "Amazon web services (aws)," https://aws.amazon.com/, 2019, accessed: 12 July 2019.

[13] Intel. (2019, September) Intel neural compute stick 2. https://software.intel.com/en-us/neural-compute-stick. Accessed: November 28, 2019.

[14] NVIDIA, "Jetson tx2 module," https://developer.nvidia.com/embedded/jetson-tx2, 2019, accessed: 12 July 2019.

[15] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[16] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

[17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. CVPR*, 2017, pp. 7310–7311.

[18] M. Almeida, S. Laskaridis, I. Leontiadis, S. I. Venieris, and N. D. Lane, "Embench: Quantifying performance variations of deep neural networks across modern commodity devices," in *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*, 2019, pp. 1–6.

[19] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware," *arXiv preprint arXiv:1812.01739*, 2018.

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

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, 2017, pp. 4278–4284.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[25] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," *arXiv preprint arXiv:1802.03268*, 2018.

[26] D. Velasco-Montero, J. Fernández-Berni, R. Carmona-Galán, and Á. Rodríguez-Vázquez, "Optimum selection of dnn model and framework for edge inference," *IEEE Access*, vol. 6, pp. 51 680–51 692, 2018.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014, pp. 580–587.

[28] R. Girshick, "Fast r-cnn," in *Proc. ICCV*, 2015, pp. 1440–1448.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[30] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. CVPR*, 2017, pp. 7263–7271.

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[32] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. CVPR*, 2018, pp. 4203–4212.

[33] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[34] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[39] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[40] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[41] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International journal of multimedia information retrieval*, vol. 7, no. 2, pp. 87–93, 2018.

[42] R. G. Trindade, J. V. F. Lima, and A. S. Charão, "Performance evaluation of deep learning frameworks over different architectures," in *International Conference on Vector and Parallel Processing*, 2018, pp. 92–104.

[43] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[44] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[45] F. Research, "Caffe2," https://caffe2.ai, September 2019, accessed: 6 September 2019.

[46] X. Zhang, Y. Wang, and W. Shi, "pcamp: Performance comparison of machine learning packages on the edges," in {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[47] D. Pena, A. Forembski, X. Xu, and D. Moloney, "Benchmarking of cnns for low-cost, low-power robotics applications," in *RSS 2017 Workshop: New Frontier for Deep Learning in Robotics*, 2017, pp. 1–5.

[48] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on android smartphones," in *Proc. ECCV*, 2018, pp. 0–0.

[49] 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*, vol. 115, no. 3, pp. 211–252, 2015.

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[52] I. Cosmic Shovel, "Amazon price tracker, amazon price history charts, price watches, and price drop alerts." https://camelcamelcamel.com/, March 2019, accessed: 24 September 2019.

[53] Intel, "Intel movidius neural compute stick," https://software.intel.com/en-us/movidius-ncs, September 2019, accessed: 6 September 2019.

[54] ——, "Openvino toolkit," https://docs.openvinotoolkit.org/2018_R5/index.html, December 2018, accessed: 3 September 2019.

[55] NVIDIA, "Accelerating inference in tf trt user guide," https://docs.nvidia.com/deeplearning/frameworks/tf-trt-user-guide/index.html, August 2019, accessed: 3 September 2019.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS Autodiff Workshop*, 2017.

[57] IBM, "Watson visual recognition: Pricing," https://www.ibm.com/cloud/watson-visual-recognition/pricing, September 2019, accessed: 13 September 2019.

Part V

Summary

Summary

The goal of this thesis is to show that vision-based person re-id can be used to measure queue times in, e.g., an airport. To accept or reject this hypothesis, this thesis has covered three important tasks within re-id; (1) data acquisition, (2) feature extraction and (3) practical re-identification.

To properly evaluate a re-id system in a context as close as possible to a real-world queue scenario, new datasets have been collected from overhead viewpoints. Due to the viewpoint of the cameras, datasets have been collected using 3D cameras that capture both RGB and depth images. Specifically, ZED cameras have been used based on their high resolution and wide field of view [1]. The first dataset, OPR, was collected from a single ZED camera at university canteen, and where persons would be captured first while standing in queue and later once they left the canteen. 64 persons were annotated and the dataset was used throughout the project to devise novel multimodal features. The second dataset, QPR, was collected in an airport from two non-overlapping ZED cameras. The goal was to collect a dataset from a real queue scenario, which could be used to both evaluate re-id precision and queue time measurement. 116 persons have been annotated and timestamps are extracted from each person.

Based on the collection of the first dataset, a CNN has been developed to learn multimodal features based on both RGB and depth image input. Experimental results show an increase in re-id precision compared to using either of the two modalities alone [2]. Next, a spatial attention module has been implemented to capture local semantics from different layers of the CNN and fuse those with global feature representations from the original work [3]. Finally, the re-id precision of the CNN has been increased even further, by adding a layer-wise attention module, which dynamically weights the local semantics extracted by the spatial attention module before they are fused with global feature representations. From the final system, we have shown state-of-the-art performance on both the novel OPR dataset and two previous datasets, DPI-T and TVPR, which are also collected from an overhead viewpoint [4]. As another important, step towards maximizing re-id precision, this thesis has also investigated late fusion of features. We have shown that

fusing features at different abstraction levels greatly increases re-id precision, both if score-level fusion or rank-aggregation is applied [5]. Due to the low computational time of late fusion, it is relevant to consider in future work.

Based on the novel features and the collected QPR dataset, we have evaluated vision-based re-id for queue measurements to answer our hypothesis. Using the existing CNN with both spatial and layer-wise attention, combined with a post processing step that performs one-to-one assignments between a set of probes and a gallery, we have shown that median queue times can be estimated that deviate only a few percentages from the ground truth. Furthermore, we have shown that one-to-one assignment between probe and gallery greatly increases re-id precision [6]. We have, thus, shown that vision-based re-id can be used to properly estimate queue times and we therefore accept our hypothesis.

Besides accepting our hypothesis, we have investigated certain challenges in deploying re-id system. First, we have shown that it is possible to transfer knowledge from an existing camera network to a newly introduced camera, without having to annotate excessive amounts of new data to properly train a new CNN model, but instead use existing distance metrics and only little newly labeled data [7]. Finally, we have investigated potential platforms to deploy such as re-id system. This has been done through an evaluation of specific edge platforms, performing common computer vision tasks. By evaluating models of different complexities, we have compared platform/model combinations to find the most optimal one based on requirements [8].

Future Work

In this thesis, we have accepted the hypothesis that vision-based re-id can used to measure queue times. Nonetheless, work is still required to deploy a robust re-id system, which continuously provides proper queue time measurements. Some of the work, which will be discussed in the following include:

- Setting up the pipeline.
- Evaluate processing time of the re-id system.
- Transfer knowledge to a new camera network.
- Evaluation of a larger dataset and across multiple days.

Setting up the Pipeline

To have a fully functional system, all steps in the re-identification pipeline (Figure 1.3 of section I.1.1) need to be implemented. An easy way is to implement a person detector, is to consider a public object detector, such as an SSD [9] or YOLO [10]. However, given the complexities of these networks, more simple ways to detect might be necessary. One way could be to use the depth and extract the background, as we did upon annotation of the QPR dataset in chapter II.3. Furthermore, a tracking algorithm is required to ensure multiple samples of each person. A simple way is to implement a Kalman filter [11], however, the challenge is to avoid id switches if multiple persons are withing the camera view.

Evaluate Processing Time

Our state-of-the-art re-id system in [4] uses a MobileNetV2 [12] as backbone. While this network is fast compared to more complex networks, such as ResNet50 [13] or InceptionV4 [14], processing time still needs to be evaluated in order to ensure that it can run on an edge platform within a specified

timing requirement. Furthermore, the complexity of the spatial and layer-wise attention modules should be analyzed to see if they can be optimized. Finally, there is also an option to use features that are less discriminative, however, faster to extract, as long as the re-id precision is within the requirement. To that end, methods, such as knowledge distillation [15], can be used to learn small networks based on knowledge from larger ones. A study of speed-precision trade-off based on the ability to correctly measure queue times is mostly relevant.

Knowledge Transfer

In [7], we have proposed a system to transfer knowledge from an existing camera network to a newly introduced camera with only limited labeled data from the expanded camera network. Another case of knowledge transfer is to learn an entire new camera network using previous knowledge with no or only few labeled data. This is a different case since new persons are only seen in new environments that may be completely different from the first one. Popular ways to deal with this type of knowledge transfer, is to perform image to image translation from the old to the new environment, to retain person id's [16, 17], or simply use a pre-trained model to add pseudo labels to persons captured in the new environment, and finetune a model based on that [18, 19].

Evaluation

In [6], we have evaluated re-id based queue time measurements using a dataset containing 90 test samples. While this gives a good initial idea of how well the system works, a more extensive test has to be conducted to validate the robustness of the system. This involves testing across several days and with a higher number of test samples.

References

[1] Stereolabs. (2017) Zed - depth sensing and camera tracking. https://www.stereolabs.com/zed/. Stereolabs. Accessed: November 28, 2019.

[2] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Multimodal neural network for overhead person re-identification," *Proc. BIOSIG*, pp. 25–34, 2017.

[3] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "Attention in multimodal neural networks for person re-identification," in *Proc. CVPR Workshops*, 2018, pp. 179–187.

[4] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1216–1231, 2019.

[5] A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Enhancing person re-identification by late fusion of low-, mid- and high-level features," *IET Biometrics*, vol. 7, no. 2, pp. 125–135, 2018.

[6] A. R. Lejbølle, B. Krogh, K. Nasrollahi, and T. B. Moeslund, "One-to-one person re-identification in a queue," Aalborg University, Tech. Rep., 2019.

[7] S. M. Ahmed, A. R. Lejbølle, R. Panda, and A. K. Roy-Chowdhury, "Camera on-boarding for person re-identification using hypothesis transfer learning," November 2019, under review for the 2020 IEEE Conference on Computer Vision and Pattern Recognition.

[8] A. R. Lejbølle, C. B. Rasmussen, K. Nasrollahi, and T. B. Moeslund, "Evaluation of edge platforms for deep learning in computer vision," under review for the journal of Neural Computing and Applications.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[10] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. CVPR*, 2017, pp. 7263–7271.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, 2017, pp. 4278–4284.

[15] I. Ruiz, B. Raducanu, R. Mehta, and J. Amores, "Optimizing speed/accuracy trade-off for person re-identification via knowledge distillation," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103309, 2020.

[16] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proc. CVPR*, June 2018, pp. 994–1003.

[17] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *Proc. CVPR*, 2018, pp. 5157–5166.

[18] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 4, p. 83, 2018.

[19] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proc. CVPR*, 2018, pp. 2275–2284.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-582-6

AALBORG UNIVERSITY PRESS