

## Computer Vision-based Monitoring of Harvest Quality

Rasmussen, Christoffer Bøgelund

*DOI (link to publication from Publisher):*  
[10.54337/aau468596652](https://doi.org/10.54337/aau468596652)

*Publication date:*  
2021

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

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
Rasmussen, C. B. (2021). *Computer Vision-based Monitoring of Harvest Quality*. Aalborg Universitetsforlag.

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### Take down policy

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.





# **COMPUTER VISION-BASED MONITORING OF HARVEST QUALITY**

**BY  
CHRISTOFFER BØGELUND RASMUSSEN**

DISSERTATION SUBMITTED 2021



**AALBORG UNIVERSITY**  
DENMARK



---

---

# Computer Vision-based Monitoring of Harvest Quality

---

---

Ph.D. Dissertation  
Christoffer Bøgelund Rasmussen

Dissertation submitted December 10, 2021

Dissertation submitted: December 10, 2021

PhD supervisor: Professor Thomas B. Moeslund  
Aalborg University

Assistant PhD supervisors: PhD Kristian Kirk  
CLAAS E-Systems  
  
B.Sc Lars Vestergaard  
CLAAS E-Systems

PhD committee: Associate Professor David Meredith (chairman)  
Aalborg University, Denmark  
  
Professor Christopher Steven McCool  
University of Bonn, Germany  
  
Senior Data Scientist, PhD: Henrik Pedersen  
Systematic, Denmark

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628

ISBN (online): 978-87-7210-926-8

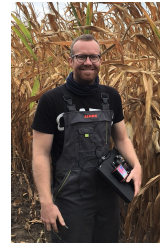
Published by:  
Aalborg University Press  
Kroghstræde 3  
DK – 9220 Aalborg Ø  
Phone: +45 99407140  
aauf@forlag.aau.dk  
forlag.aau.dk

© Copyright: Christoffer Bøgelund Rasmussen

Printed in Denmark by Rosendahls, 2022

# Curriculum Vitae

Christoffer Bøgelund Rasmussen



Christoffer Bøgelund Rasmussen received his M.Sc in Vision, Graphics and Interactive Systems in 2017 from Aalborg University, Denmark. Following the completion of his M.Sc he joined CLAAS E-Systems as a computer vision engineer and started his PhD project after a successful application for funding through Innovation Fund Denmark.

His main research interests include computer vision and machine learning, particularly with deep neural networks. Furthermore, he has an interest in applying such research areas into the real-world with a focus on business value. As part of his PhD he has also been involved in supervision and teaching of graduate and undergraduate students.

## Curriculum Vitae

# Abstract

An efficient and robust method to measure the quality of harvested corn silage is essential for a farmer in order to optimise yield and machine efficacy. Current measurement approaches are cumbersome and time-consuming due to manual sample preparation and separation steps.

This PhD thesis investigated automatic monitoring of corn silage quality from images taken directly after harvesting, without the need for manual processing such as separating particles. Concretely, we proposed to use deep learning for localising kernel fragments and stover overlengths. Two-stage networks were evaluated and improved upon in comparison to a more naive training approach. This was realised through investigations in data sampling, finetuning, and adapting priors for object shape and size when initialising training. In addition to two-stage networks, a novel network was presented aimed to efficiently classify sieve sizes for kernel fragments without the need for regression or classification modules. The proposed networks showed promising results when evaluated with annotations or correlation to physically sieved samples.

As deep learning and neural networks played a central role in this thesis, investigations were made into understanding the models. A number of deep learning models were evaluated for kernel fragment recognition in relation to precision, recall and speed, and from this, the optimal deep learning architecture was proposed. Deep learning models require a large amount of high quality annotated data for development and evaluation. Larger datasets achieve this through extensive pipelines, however, this can be expensive to implement in more fine-grain tasks. Therefore, we covered the challenges in annotating corn silage and investigated alternatives to manual annotation. Finally, deploying models for different computer vision tasks on edge devices was investigated in regards to the trade-offs in speed and retail price.

This work has covered methodologies for the potential of an automatic, efficient and robust monitoring of corn silage quality, while covering the real-world aspects of working with deep learning. Hereby paving the way for automated optimal machine settings during harvesting, which can ensure corn silage of high quality.

## Abstract



# Resumé

En effektiv og robust metode til at måle kvaliteten af høstet majsensilage er afgørende for at landmanden kan optimere udbytte og maskineffektivitet. Dog er nuværende målemetoder er besværlige og tidskrævende på grund af manuel prøveforberedelse og separationstrin.

Denne ph.d.-afhandling undersøgte automatisk overvågning af kvaliteten af majsensilage fra billeder taget direkte af planten efter høst uden brug af adskillelsestrin. Konkret foreslog vi at bruge *deep learning* til at lokalisere kernefragmenter og overlængder af majsblade. To-trins netværk blev evalueret og forbedret i forhold til en mere naiv træningstilgang. Dette blev realiseret gennem undersøgelser i datasampling, finjustering og tilpasning af forudsætninger for objektets form og størrelse ved træningens start. I tillæg til to-trins netværk blev et nyt netværk præsenteret med det formål at klassificere sigtestørrelser for kernefragmenter uden behov for regression eller klassifikationsmoduler. Alle netværk viste lovende resultater, når de blev evalueret med annoteringer og/eller korrelation til fysisk sigtede prøver.

Da *deep learning* og neurale netværk spillede en central rolle i denne afhandling, blev der foretaget undersøgelser af forståelsen af modellerne. En række *deep learning* modeller til kernefragmentgenkendelse blev evalueret i forhold til præcision, recall og hastighed, og ud fra dette blev den optimale *deep learning* arkitektur foreslået. *Deep learning* modeller kræver en stor mængde annoterede data af høj kvalitet til udvikling og evaluering. Større datasæt opnår dette gennem omfattende pipelines, men dette kan være dyrt at implementere i mere specifikke opgaver. Derfor diskuterede vi udfordringerne i at annotere majsensilage data og undersøgte alternativer til manuel annotering. Endelig blev implementeringen af modeller til forskellige computervisionsopgaver på *edge*-enheder undersøgt med hensyn til kompromis mellem hastighed og detailpris.

Dette arbejde har muliggjort udviklingen af metoder til en automatisk, effektiv og robust overvågning af majsensilagekvalitet, hvor der er taget højde for de praktiske aspekter ved at arbejde med *deep learning*. Dette er med til at bane vejen for optimale maskinindstillinger, som sikrer majsensilage af højeste kvalitet.

## Resumé

# Contents

<b>Curriculum Vitae</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Resumé</b>	<b>vii</b>
<b>Thesis Details</b>	<b>xiii</b>
<b>Preface</b>	<b>xvii</b>
<b>List of Abbreviations</b>	<b>xix</b>
<b>I Overview</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1 Monitoring Corn Silage Quality . . . . .	4
2 Understanding Neural Networks . . . . .	7
3 Thesis Structure . . . . .	9
References . . . . .	9
<b>2 Monitoring Corn Silage Quality</b>	<b>15</b>
1 Introduction . . . . .	15
2 State-of-the-art . . . . .	17
3 Contributions . . . . .	22
References . . . . .	28
<b>3 Understanding Neural Networks</b>	<b>39</b>
1 Introduction . . . . .	39
2 State-of-the-art . . . . .	41
3 Contributions . . . . .	46
References . . . . .	49

<b>4</b>	<b>Conclusion</b>	<b>57</b>
<b>II</b>	<b>Monitoring Corn Silage Quality</b>	<b>63</b>
<b>A</b>	<b>Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Images</b>	<b>65</b>
1	Introduction . . . . .	67
2	Materials and Methods . . . . .	70
3	Results . . . . .	78
4	Discussion . . . . .	87
5	Conclusions . . . . .	88
	References . . . . .	89
<b>B</b>	<b>Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics</b>	<b>93</b>
1	Introduction . . . . .	95
2	Related Work . . . . .	98
3	Methodology . . . . .	100
4	Results . . . . .	106
5	Discussion . . . . .	118
6	Conclusion . . . . .	120
7	Appendix . . . . .	121
	References . . . . .	132
<b>C</b>	<b>SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage</b>	<b>137</b>
1	Introduction . . . . .	139
2	Related Work . . . . .	141
3	Methodology . . . . .	143
4	Results . . . . .	146
5	Conclusion . . . . .	150
	References . . . . .	151
<b>III</b>	<b>Understanding Neural Networks</b>	<b>153</b>
<b>D</b>	<b>Evaluation of Model Selection for Kernel Fragment Recognition in Corn Silage</b>	<b>155</b>
1	Introduction . . . . .	157
2	Data . . . . .	158
3	CNN Meta-Architectures . . . . .	158
4	Results . . . . .	159

## Contents

5	Conclusions . . . . .	161
	References . . . . .	163
<b>E</b>	<b>The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage</b>	<b>165</b>
1	Introduction . . . . .	167
2	Related Work . . . . .	169
3	Dataset Annotation . . . . .	172
4	Semi-Supervised Learning . . . . .	182
5	Discussion . . . . .	184
6	Conclusion . . . . .	185
	References . . . . .	187
<b>F</b>	<b>Evaluation of Edge Platforms for Deep Learning in Computer Vision</b>	<b>191</b>
1	Introduction . . . . .	193
2	Related Work . . . . .	194
3	Platform Evaluation . . . . .	196
4	Experimental Results . . . . .	199
5	Conclusion . . . . .	204
	References . . . . .	205

## Contents

# Thesis Details

**Thesis Title:** Computer Vision-based Monitoring of Harvest Quality  
**Ph.D. Student:** Christoffer Bøgelund Rasmussen  
**Supervisors:** Professor Thomas B. Moeslund, Aalborg University  
PhD Kristian Kirk, CLAAS E-Systems  
B.Sc. Lars Vestergaard, CLAAS E-Systems

Part I of this thesis consists of an overview and contributions into the topics of Monitoring Corn Silage Quality and Understanding Neural Networks. Part II and III of this thesis contains the following papers within the two topics. All are published or accepted except [E], which is in review.

## Monitoring Corn Silage Quality

- [A] Christoffer Bøgelund Rasmussen and Thomas B. Moeslund, "Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Images," *MDPI Sensors*, Vol. 19(16), pp. 3506, 2019.
- [B] Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B. Moeslund, "Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics," *Elsevier Computers and Electronics in Agriculture* Vol. 188, pp. 106344, 2021.
- [C] Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B. Moeslund, "SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage," *Accepted as a full paper at the 17th International Conference on Computer Vision Theory and Applications (VISAPP) 2022*.

## Understanding Neural Networks

- [D] Christoffer Bøgelund Rasmussen and Thomas B. Moeslund, "Evaluation of Model Selection for Kernel Fragment Recognition in Corn Silage," *ICLR 2020 Workshop on Computer Vision for Agriculture (CV4A) - Virtual*, 2020.
- [E] Christoffer Bøgelund Rasmussen, Kristian Kirk, and Thomas B. Moeslund, "The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage," *Submitted to MDPI Sensors*, 2021.
- [F] Christoffer Bøgelund Rasmussen, Aske Rasch Lejbølle, Kamal Nasrollahi and Thomas B. Moeslund "Evaluation of Edge Platforms for Deep Learning in Computer Vision," *In: Del Bimbo A. et al. (eds) Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science, vol 12664. Springer, pp. 523-537, 2021.*

Furthermore, as part of the work conducted during this PhD the following two patents are submitted to the patent offices of Europe and United States of America related to Monitoring Corn Silage Quality. The patents have the same title but different focus on novelty. Specifically, one patent for the algorithm and another for application of the method with a harvester.

- Christoffer Bøgelund Rasmussen, Kristian Kirk, Thomas B. Moeslund, Sven Carsten Belau and Frédéric Fischer, "Method for detecting lengths of a particle," *German patent no. DE10 2021 105 274.2*, 2021.
- Christoffer Bøgelund Rasmussen, Kristian Kirk, Thomas B. Moeslund, Sven Carsten Belau and Frédéric Fischer, "Method for detecting lengths of a particle," *German patent no. DE10 2021 105 273.4*, 2021.

In addition to the above papers, the following publications have also been co-authored but are not in consideration for this PhD.

- Amélie Beucher, Christoffer Bøgelund Rasmussen, Mogens Greve and Thomas B. Moeslund, "Interpretation of Convolutional Neural Networks for Acid Sulfate Soil Classification," *Accepted/in press to Frontiers in Environmental Science*, 2021.
- Christoffer Bøgelund Rasmussen, Kamal Nasrollahi and Thomas B. Moeslund, "R-FCN Object Detection Ensemble based on Object Resolution and Image Quality," *International Joint Conference on Computational Intelligence*, pp. 110-120, 2017.
- Andreas Aakerberg, Kamal Nasrollahi, Christoffer Bøgelund Rasmussen and Thomas B. Moeslund, "Depth Value Pre-processing for Accurate



Transfer Learning Based RGB-D Object Recognition," *International Joint Conference on Computational Intelligence*, pp. 121-128, 2017.

- Daniel Kold Hansen, Kamal Nasrollahi, Christoffer Bøgelund Rasmussen and Thomas B. Moeslund, "Real-time Barcode Detection and Classification Using Deep Learning," *International Joint Conference on Computational Intelligence*, pp. 321-327, 2017.
- Andreas Aakerberg, Christoffer Bøgelund Rasmussen, Kamal Nasrollahi and Thomas B. Moeslund, "Complementing SRCNN by Transformed Self-Exemplars," In: Nasrollahi K. et al. (eds) *Video Analytics. Face and Facial Expression Recognition and Audience Measurement. VAAM 2016, FFER 2016. Lecture Notes in Computer Science*, vol 10165. Springer, Cham., pp. 127-136, 2016.

## Thesis Details

# Preface

This Industrial PhD study is a collaboration between the Visual Analysis and Perception (VAP) laboratory at the Section of Media Technology, Aalborg University and CLAAS E-Systems Denmark, who have developed innovative computer vision solutions for agriculture for over 20 years. The thesis covers the themes of monitoring corn silage quality and understanding neural networks. Part one introduces the topics, provides an overview of state-of-the-art and the contributions to the fields. Following this, part two and three consists of papers within the two themes. Thus, this thesis is submitted as a collection of papers in partial fulfilment of a PhD study at the Section of Media Technology, Aalborg University, Denmark.

I would like to thank my supervisor, Professor Thomas B. Moeslund for giving me the opportunity to pursue the PhD together with your good supervision and motivation. Thank you to my colleagues at VAP for making me feel welcome whenever I took the trip from Copenhagen. Also, thanks to Kamal Nasrollahi for your supervision during my master's education and introducing me to deep learning. I would also like to thank the CLAAS group for allowing me the opportunity to pursue a PhD and follow my interests within your products. Thank you to my colleagues at CLAAS E-Systems Denmark for making me feel part of the team and aiding me in my work wherever possible. Thank you to my main company supervisor Kristian Kirk for your supervision and always taking the time whenever I asked. Also, thank you to Allan Kildeby for the support and allowing me to focus during the final stages of my thesis.

Thank you to my family for for always being there. Especially, to my mother Trine who helped me take the step in moving from Australia almost 10 years ago to pursue a new life. Last but not least, thank you to Pernille for your love and support. You have been amazing over the past months when I have been immersed in my work. I look forward to everything the future brings for us.

Christoffer Bøgelund Rasmussen  
Aalborg University, December 10, 2021

## Preface

# List of Abbreviations

AI	Artificial Intelligence
AMT	Amazon Mechanical Turk
ASABE	American Society of Agricultural and Biological Engineers
AWS	Amazon Web Services
AP	Average Precision
AR	Average Recall
CCC	Concordance Correlation Coefficient
CL	Cutting Length
CV	Computer Vision
CW	Calender Week
CNN	Convolutional Neural Network
CSPS	Corn Silage Processing Score
DNN	Deep Neural Network
EMA	Exponential Moving Average
FC	Fully-Connected
FCN	Fully Convolutional Network
FPS	Frames Per Second
FRCNN	Faster R-CNN
GFLOPS	Giga Floating Point Operations
GPU	Graphics Processing Unit
IQA	Image Quality Assessment
Iv2	Inceptionv2
IoA	Intersection-over-Area
IoU	Intersection-over-Union
IR	Intermediate Representation
KPS	Kernel Processing Score
MNC	Multi-task Network Cascade
MRCNN	Mask R-CNN
NCS	Neural Compute Stick
NIRS	Near-Infrared Spectroscopy
NMS	Non-Maximum Suppression

## List of Abbreviations

OVPS	Overlength Particle Score
peNDF	physically effective Neutral Detergent Fibre
PCC	Pearson Correlation Coefficient
PG	Processor Gap
PSD	Particle Size Distribution
PSPS	Penn State Particle Separator
R-FCN	Region-based Fully Convolutional Network
RGB	Red Green Blue
RMSE	Root Mean Square Error
RoI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SSD	Single Shot Multibox Detector
SSL	Semi-Supervised Learning
SVM	Support Vector Machine
TDP	Thermal Design Power
TLOC	Theoretical Length of Cut
WPCS	Whole Plant Corn Silage

# **Part I**

## **Overview**





# Chapter 1

## Introduction

Many parts of industry are becoming increasingly more automated, including agriculture. A significant driving force can be due to projected food demands as the global population is predicted to increase to over 9 billion by 2050 [1]. It is estimated that food production must increase at even higher rates due to current trends in eating habits, in addition, income inequality is reducing significantly between countries. Based upon these trends, by 2050 food production must be increased by 70%, however, this challenge is not easily solved as most of the land suitable for farming is already in use [1]. Suggestions to solving these problems include investing in technologies that lead to higher yields, in addition, the technologies should improve current food production as it is considered largely unsustainable in terms of negative effects on the climate [2].

Agriculture is one of the cornerstones of our society and there has been a countless number of technological advancements over many centuries. Major improvements in recent generations include precise in-field localisation with global navigation systems [3] or the use of bioengineering in order to create crops with enhanced genetics [4]. Current trends include big data as the number of sensors increase together with improved infrastructure for transferring data in the field [5]. Finally, image sensors is an area of research that is especially active ranging from satellite imagery, thermal, hyper-spectral and traditional RGB cameras [6].

Many occupations in agriculture require intensive work hours and automating some of the processes would provide an aid to farmers. One such area is monitoring the quality of crop harvested by a harvester. A farmer spends a large amount of time before harvesting their crop, including soil preparation, sowing seeds and weed management. Therefore, finding the correct settings for the machine is essential in order to optimise yield and profits.

This leads to the overall research question on whether the harvest quality monitoring can be automated using deep learning algorithms on RGB images. Secondly, since this is an Industrial PhD study a secondary research question arises in which practical implications this may have. Concretely, this PhD addresses monitoring the quality of corn silage harvested with forage harvesters where there is minimal efficient and effective methods currently available to the farmer. By following the recent advancements in neural networks and big data we also adopt methods that can be considered as black-boxes in this PhD, therefore, we also cover investigating and understanding neural networks.

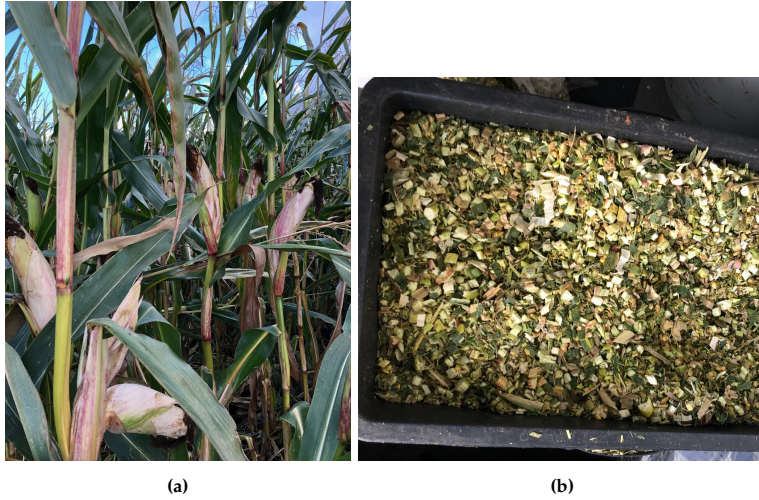
## 1 Monitoring Corn Silage Quality

Determining the quality of harvested Whole Plant Corn Silage (WPCS) is an essential step for a farmer. WPCS is a popular form of fodder for dairy cows as it can be efficiently harvested and can provide high amounts of nutrient energy [7]. The plant has a high starch content present in the corn kernels but it is protected by the outer shell, therefore, the kernels must be fragmented to expose the endosperm allowing for a more effective fodder [8]. The stover (leaves and stalks) are chopped into particles with an aim of producing physically effective Neutral Detergent Fibre (peNDF). A correct particle length increasing the peNDF is key as these are directly related to each other resulting in higher quality silage [9]. Longer particles leading to peNDF is desirable as it promotes a healthy rumen due to increased cud chewing, increased time of particles in the rumen and an overall healthy pH [10–12]. However, if particles are too long the silage can be difficult to pack tightly, potentially promoting unwanted bacterial growth [11, 13]. Furthermore, longer particles can result in extended eating times and cows may sort the feed such that they target smaller particles (i.e. kernels) [11, 12].

In Figure 1.1(a) an image is shown of the corn plant with cobs, protected by husks, attached to the stalk of the plant with its accompanying leaves. The entire plant is harvested in WPCS and an example is shown in Figure 1.1(b).

There are a number of options that a farmer can adjust when aiming for high quality WPCS and in this work we focus on two key machine settings, namely, the Processor Gap (PG) and Theoretical Length of Cut (TLOC). Figure 1.2 shows these main parts in relation to each other in a forage harvester. First, the corn plant is fed into the machine (1) to a drum of rotating knives (2), the knife drum rotates and cuts the plant according to the calibrated speed via the TLOC, after which two rotating processor rolls compresses the plant and fragment the kernels based on the PG (3). Lastly, an accelerator (4) passes the plant via a spout to be dispensed into an external trailer. Depending on the processor rolls and knife drum used in the machine, modern

## 1. Monitoring Corn Silage Quality



**Fig. 1.1:** The entire corn plant with corn cobs, stalks and leaves (a) is harvested by fragmenting kernels and chopping stover resulting in WPCS (b).

forage harvesters can set the PG between 3 to 30 mm and the TLOC can range between 4 to 44 mm.



**Fig. 1.2:** Overview of inside a forage harvester. Corn plant is fed into the machine (1), chopped by the knife drum (2), cracked by processor rolls (3) and accelerated to the spout (4) © CLAAS.

These two settings should be monitored regularly as variations within a number of factors can alter the kernel processing and peNDF. For example, in

more mature plants with a higher dry matter content, the processor rolls can be less effective as kernels become harder and more difficult to break [7]. In addition, decreasing sharpness of the knives in the rotating drum over a harvest season can shear and slip on the plant [14]. The structure of a farm can also influence how a farmer decides to adjust their harvesting settings. For example, the final particle size of WPCS can be affected differently depending on if the silage is stored in an upright silo, with a silage bag system, or in a drive-over pile bunker [12]. Additionally, when the WPCS is transported to the feeding site, the particle size can be further reduced [12]. Finally, a modern forage harvester can harvest multiple tonnes per hour using up to 180 litres of fuel [15], therefore, suboptimal settings leading to overuse of the machine can result in losses from incorrectly harvested crop and unnecessary machine wear.

When the WPCS is used as feed for dairy cows there are recommendations that farmers can follow to increase their feed quality. Kernels should be harvested when the moisture content is between 55 to 70% and only 3 to 8% of stover particles should remain when passing through a 1.9 cm sieve [16]. A farmer typically measures the quality of their WPCS based upon physical measurement with sieving systems. There are options that can be used in the field, such as the Penn State Particle Separator (PSPS) [13] shown in Figure 1.3, that requires an operator to manually shake a number of sieves from which the Particle Size Distribution (PSD) can be measured. However, as this must be done manually the process can be cumbersome and prone to error. Off-site options also exist where a WPCS sample is sent for an in-depth analysis, such as determining the Corn Silage Processing Score (CSPS) [17] or the American Society of Agricultural and Biological Engineers (ASABE) particle separator [18]. However, the analysis can take multiple days making it impossible for a farmer to adjust their machine during harvesting. An optimal system for determining the machine efficacy and quality of harvested silage would occur in the field efficiently and without the need for manual steps.



**Fig. 1.3:** PSPS separates WPCS using stacked sieves which an operator has to manually shake. The distribution of particles on the sieves can describe the quality. Image from [19] © CLAAS.

## 2 Understanding Neural Networks

In recent years, neural networks have seen significant growth and adaptation in both research and industry. This can be attributed to impressive results on benchmark challenges, such as in 2012 when AlexNet greatly improved the classification accuracy on ImageNet [20] and Artificial Intelligence (AI)-based systems surpassing humans in a number of applications such as cancer detection [21]. In academia, the growth can be seen in the increase of papers, for example, on arXiv in recent years as shown in Figure 1.4, where the number has especially grown in the field of machine learning and computer vision and pattern recognition [22].

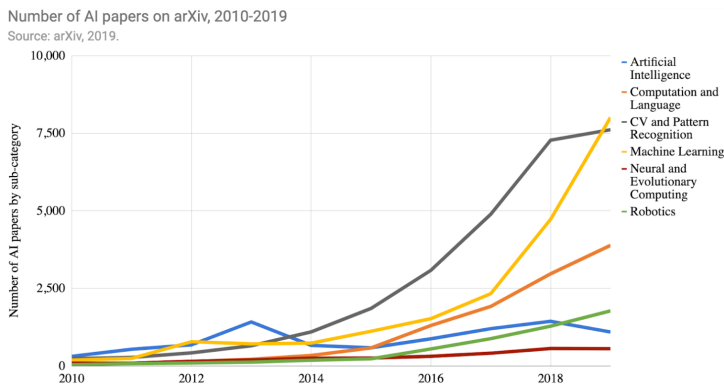


Fig. 1.4: AI related publications on arXiv. Image from [22] © Stanford University.

Whereas in industry the global investment has increased significantly, by the order of billions as seen in Figure 1.5 for AI startups [22].

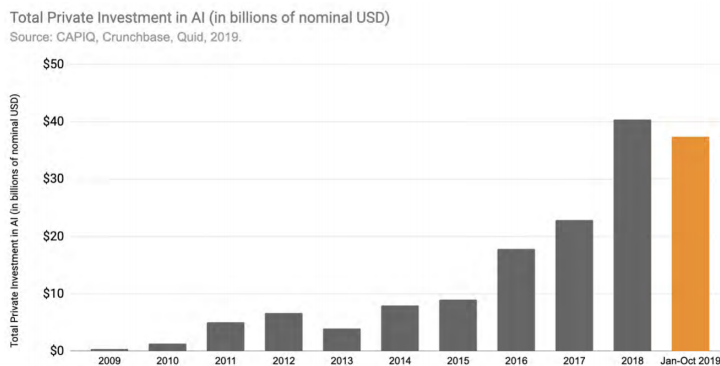
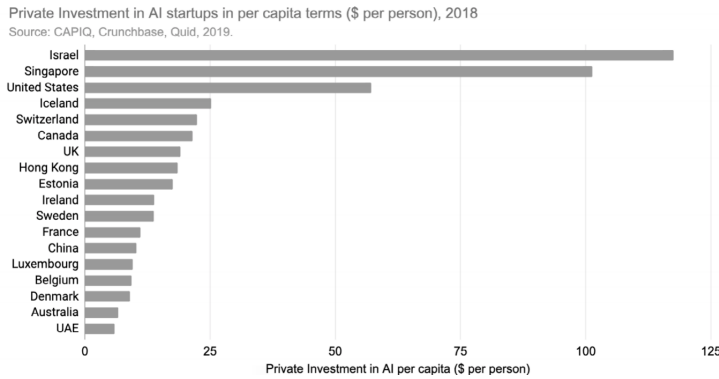


Fig. 1.5: Global investment in AI startups. Image from [22] © Stanford University.

AI and neural networks have great potential in Industry 4.0 and enabling the technology has become a strategy for a number of nations, including Denmark. While Denmark is competitive in AI it is still behind other countries, as seen in Figure 1.6, ranking lower when measuring investments based on gross domestic product, especially compared to Israel, Singapore and the United States [22].



**Fig. 1.6:** Investment in AI startups ranked by gross domestic product. Image from [22] © *Stanford University*.

In response to such statistics the Danish government published a report in 2019 outlining the strategy for AI. The report states that agriculture is one of four focus areas that will be prioritised for gaining AI experience [23]. Precision agriculture is mentioned as a case where AI, together with high quality data sources, can strengthen the Danish economy and have a positive effect on the climate [23]. This can be further highlighted by reports that AI in agriculture is currently valued at USD\$ 608.9 million and expected to grow annually by 25.4% between 2019 and 2025 to USD\$ 2.9 billion [24, 25].

In the Danish National Strategy for Artificial Intelligence the government defines four key objectives. These are quoted here:

1. "Denmark should have a common ethical and human-centred basis for artificial intelligence,
2. Danish researchers should research and develop artificial intelligence,
3. Danish businesses should achieve growth through developing and using artificial intelligence and
4. The public sector should use artificial intelligence to offer world-class services" [23].

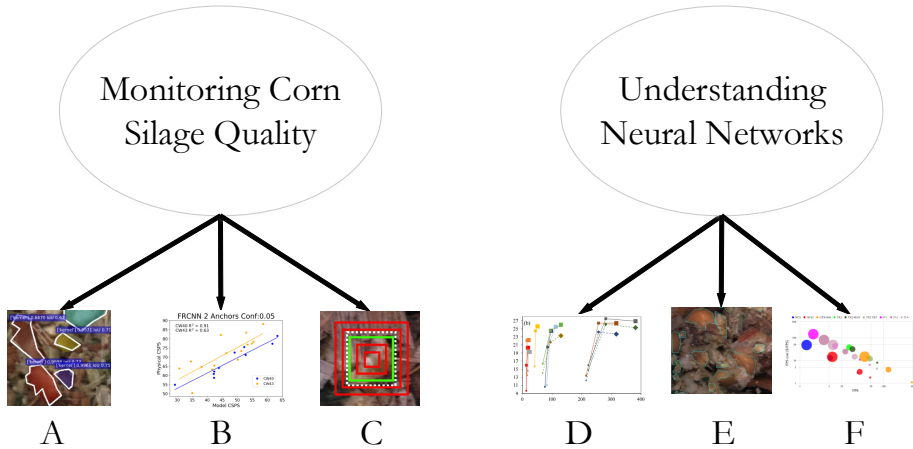
By researching the use of neural networks for monitoring WPCS harvesting quality this work will cover objectives 2 and 3, furthermore, we aim to

investigate models in real-world scenarios. A key item the report mentions are challenges in moving towards AI systems related to transparency when working with large amounts of data and complex models [23]. Therefore, in this PhD there is an aim to have a focus on the understanding of neural networks.

## 3 Thesis Structure

This PhD will give an overview to monitoring corn silage quality and understanding neural networks. The two themes will include an overview of state-of-the-art and the contributions to the respective fields in Part I.

Following the overview will be an appendix of two parts, consisting of a collection of papers for the two topics. Part II will cover the work on monitoring corn silage quality where we investigate the usage of deep learning for localising objects relevant to WPCS harvest quality. Part III covers the work done on understanding neural networks in regards to deployability in real-world scenarios and dataset creation. As illustrated in Figure 1.7, the collection of papers in the appendix will include three works on monitoring corn silage quality and three on understanding neural networks.



**Fig. 1.7:** The thesis is structured as a collection of papers within monitoring corn silage quality and understanding neural networks. Figure adapted with images from [26–31].

## References

- [1] FAO, “How to Feed the World 2050,” [http://www.fao.org/fileadmin/templates/wsfs/docs/expert\\_paper/How\\_to\\_Feed\\_the\\_World\\_in\\_2050.pdf](http://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf) (accessed February 10, 2020), 2009.

## References

- [2] Foresight, "The Future of Food and Farming (2011) Final Project Report," [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/288329/11-546-future-of-food-and-farming-report.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/288329/11-546-future-of-food-and-farming-report.pdf) (accessed February 21, 2020), 2011.
- [3] A. ur Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, "A review of wireless sensors and networks' applications in agriculture," *Computer standards and interfaces*, vol. 36, no. 2, pp. 263–270, 2014.
- [4] A. C. Tyagi, "Towards a second green revolution," *Irrigation and Drainage*, vol. 65, no. 4, pp. 388–389, 2016. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ird.2076>
- [5] A. Kamilaris, A. Kartakoullis, and F. X. Prenafeta-Boldú, "A review on the practice of big data analysis in agriculture," *Computers and Electronics in Agriculture*, vol. 143, pp. 23–37, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169917301230>
- [6] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169917308803>
- [7] L. Ferraretto, R. Shaver, and B. Luck, "Silage review: Recent advances and future technologies for whole-plant and fractionated corn silage harvesting," *Journal of Dairy Science*, vol. 101, no. 5, pp. 3937–3951, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030218303199>
- [8] T. McAllister, H. Bae, G. Jones, and K. Cheng, "Microbial attachment and feed digestion in the rumen," *Journal of Animal Science*, vol. 72, no. 11, pp. 3004–3018, 1994.
- [9] D. Mertens, "Creating a system for meeting the fiber requirements of dairy cows," *Journal of Dairy Science*, vol. 80, no. 7, pp. 1463–1481, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030297760752>
- [10] L. F. Ferraretto and R. D. Shaver, "Meta-analysis: Effect of corn silage harvest practices on intake, digestion, and milk production by dairy cows," *The Professional Animal Scientist*, vol. 28, pp. 141–149, 2012.
- [11] J. Townsend, "Dairy cattle stresses caused by forages what i have seen and how to avoid them," 2005.
- [12] D. Wiersma, "Theoretical length of cut: Theory and practice," <https://www.progressiveforage.com/forage-production/>



## References

- management/theoretical-length-of-cut-theory-and-practice, 2013, accessed: 08 December 2021.
- [13] J. Heinrichs and M. J. Coleen, "Penn state particle separator," May 2016. [Online]. Available: <https://extension.psu.edu/penn-state-particle-separator>
- [14] K. Shinnars, "Engineering principles of silage harvesting equipment," *Silage Science and Technology*, pp. 361–403, 2003.
- [15] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [16] Heinrichs, J. and Ishler, V.A and Roth, G.W., "From Harvest to Feed: Understanding Silage Management," <https://extension.psu.edu/from-harvest-to-feed-understanding-silage-management> (accessed March 4, 2021), 2017.
- [17] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," *In: Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [18] ASABE, "Method of determining and expressing particle size of chopped forage materials by screening," *ANSI/ASAE*, vol. S424.1, p. 663–665.
- [19] CLAAS, "Shredlage harvest recommendations," <https://www.claas.co.uk/products/technologies/shredlage/harvest-recommendations>, 2021, accessed: 19 March 2021.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [21] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F. J. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C. J. Kelly, D. King, J. R. Ledsam, D. Melnick, H. Mostofi, L. Peng, J. J. Reicher, B. Romera-Paredes, R. Sidebottom, M. Suleyman, D. Tse, K. C. Young, J. D. Fauw, and S. Shetty, "International evaluation of an AI system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.

## References

- [22] S. University, "The 2019 ai index report," [https://hai.stanford.edu/sites/default/files/ai\\_index\\_2019\\_report.pdf](https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf), 2019, accessed: 19 March 2021.
- [23] Ministry of Finance and Ministry of Industry, Business and Financial Affairs, "National Strategy for Artificial Intelligence," [https://en.digst.dk/media/19337/305755\\_gb\\_version\\_final-a.pdf](https://en.digst.dk/media/19337/305755_gb_version_final-a.pdf) (accessed March 4, 2021), 2019.
- [24] Bloomberg, "Artificial intelligence in agriculture market," <https://www.bloomberg.com/press-releases/2020-01-08/artificial-intelligence-in-agriculture-market-size-worth-2-9-billion-by-2025-cagr-25-4-grand-view-research-inc>, 2020, accessed: 19 March 2021.
- [25] G. V. Research, "Artificial intelligence in agriculture market size, share & trends analysis report by component (software, hardware), by technology, by application (precision farming, drone analytics), by region, and segment forecasts, 2019 - 2025," <https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-in-agriculture-market>, 2019, accessed: 19 March 2021.
- [26] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [27] C. B. Rasmussen and T. B. Moeslund, "Evaluation of model selection for kernel fragment recognition in corn silage," <https://arxiv.org/abs/2004.00292>, 2020.
- [28] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Anchor tuning in faster r-cnn for measuring corn silage physical characteristics," *Computers and Electronics in Agriculture*, vol. 188, p. 106344, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169921003616>
- [29] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Sievenet: Estimating the particle size distribution of kernel fragments in whole plant corn silage," *Accepted/in press at the 17th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022.
- [30] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "The challenge of data annotation in deep learning – a case study on whole plant corn silage," *Under review at MDPI Sensors*, 2021.
- [31] C. B. Rasmussen, A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Evaluation of edge platforms for deep learning in computer

## References

vision,” in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds. Cham: Springer International Publishing, 2021, pp. 523–537.

## References

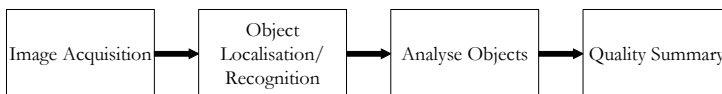
## Chapter 2

# Monitoring Corn Silage Quality

### 1 Introduction

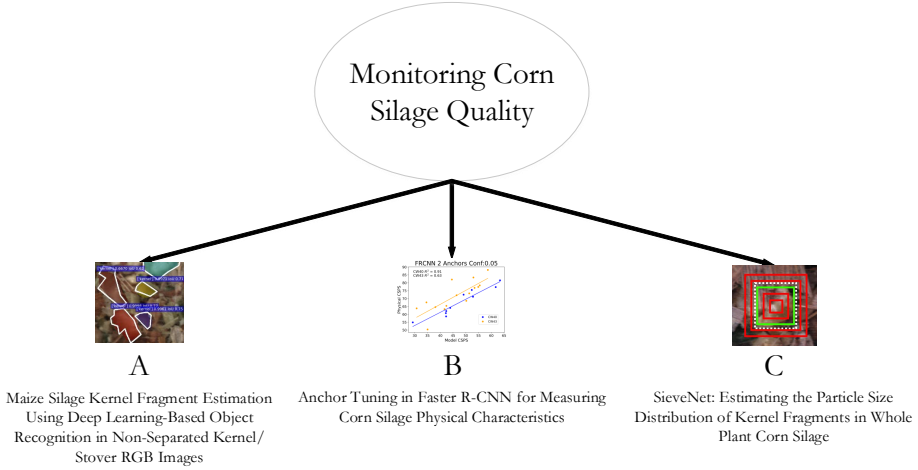
Computer vision systems are abundant in industry and can lower the requirements on human analysis where tasks can be difficult or tedious to perform. As mentioned earlier, an efficient quality monitoring system can have significant impact on WPCS harvesting as current analysis requires human knowledge, cumbersome manual preparation and in some cases to be sent to an off-site laboratory. We propose solving this problem with the usage of computer vision and neural networks to measure the WPCS quality using metrics that can describe the shape and size of the harvested crop.

To give an overview of how computer vision could solve this task we show the general steps for a typical industrial vision system based on object recognition in Figure 2.1. First, an image is acquired from a sensor, such as an RGB camera, together with illumination that allows the sensor to capture the reflected light from the scene. Following this, object recognition can provide localisation and categorisation. This final module is built by defined rules utilising features with image processing techniques or learnt via machine learning. Third, the objects are analysed, for example, by shape and size characteristics. Finally, a summary of the quality in the scene can be made for a single or group of images.



**Fig. 2.1:** Simplified industrial computer vision system based on object recognition.

To understand the opportunities and challenges for creating a computer vision system for WPCS an overview into the current state-of-the-art for monitoring harvest quality will be provided. From this, three works completed as part of the PhD to monitoring corn silage quality will be presented, as shown in Figure 2.2, together with a number of contributions. However, before the state-of-the-art will be presented it is appropriate to explain in more detail the current methodologies used on the farm.



**Fig. 2.2:** Overview of the works presented in this thesis for monitoring corn silage quality. Figure with images from [1–3].

## 1.1 Current Practices

### Kernel Processing

As covered earlier, two of the major settings a farmer can adjust during harvesting is the PG and TLOC. Both settings affect the fragmentation of kernels and chopping of the stover, however, primarily the PG is related to the kernel processing and TLOC to the stover. Current methods require the separation of the kernels and stover as this simplifies the task and can provide a basis for an accurate measurement. An example of a fast but subjective method is by hydrodynamic separation [4], here, a sample of WPCS is placed in water and stirred until kernel fragments sink and stover float due to the lower buoyancy of the kernels. Once separated, the kernel processing can be subjectively determined. The industry accepted standard for measuring kernel processing is the CSPS [5]. This method takes a sample of WPCS and passes it through a number of sieves to separate the stover and kernels. To determine the CSPS, the percentage of kernel fragments passing a 4.75 mm sieve is measured for a sample. With CSPS, the authors defined that above 70% of

fragments passing shows optimal processing, between 50 to 60% is sufficient and below 50% is suboptimal. However, there have been criticisms to this approach as it requires the assumption that all fragments passing through the sieve are of equal quality and this is not necessarily the case. For example, particles that can pass a 1.18 mm sieve are ineffective as fodder as they rapidly pass through a cow's rumen [6]. Furthermore, in [7] the authors determined that the starch content varied among the PSD of a kernel fragment sample and using a single sieve to describe the quality was not adequate for feeding models. Therefore, the authors determined that estimating the geometric mean particle size by first using hydrodynamic separation [4] followed by sieving dried samples gave a better description of the kernel processing compared to CSPS.

### **Stover Chopping**

For stover particles, typically the aim is to estimate the peNDF which is done through size characteristics, such as mean particle length [8]. A popular in-field method is to manually shake a number of stacked sieves, defined in [9], separating the stover. Since the original stacked system, a number of sieves have been replaced as the understanding of how PSD affects the digestion has progressed and now the PSPS is commonly used [10]. The separator has a relatively specific set of instructions for operation, that includes a total of eight iterations of shaking the sieves 5 times, where the sieves are rotated a quarter turn between each iteration. There are recommendations for the force of shaking, however, it has been argued that it can be difficult to maintain consistency and potentially lead to errors in the measurement [11]. Options also exist for off-site laboratory sieving, where the standard is the ASABE separator [12]. However, the time from harvest to measurement can be long and the machinery is considerably large weighing above 200 kg.

## **2 State-of-the-art**

This section will first cover the state-of-the-art for measuring WPCS from a computer vision perspective with respect to kernel fragments and stover particles. As the works on WPCS with computer vision is limited, we also look into a broader context with respect to other agricultural crops and determining the PSD in other domains.

### **2.1 Monitoring Corn Silage Quality**

As mentioned, the works within measuring WPCS with computer vision are limited, additionally, they all require a sample preparation step to separate

kernel and stover particles.

First, in [13], the authors aim to estimate CSPS in kernels separated from a WPCS sample and spread out on a black background. An example of the required sample preparation is shown in Figure 2.3.



**Fig. 2.3:** Before performing computer vision analysis in [13], kernel fragments must be separated from stover particles and spread out a black background together with a coin for size reference.

Fragments are found through thresholding a grayscale image where the optimal threshold is found with Maximally Stable External Regions. Kernel contours can then be found and the diameter of the maximum inscribed circle is compared against the CSPS 4.75 mm threshold from [5]. Additionally, as shown in Figure 2.3, a coin must be placed in the image as a reference to convert pixels to mm. The authors found a strong correlation between their method on images with accompanied measured CSPS, with a Pearson Correlation Coefficient (PCC) of 0.8 over 23 samples. From this work, an accompanying smartphone application named SilageSnap was developed for both Android and iOS [14]. Furthermore in [15], SilageSnap has been used to estimate CSPS in samples of WPCS with disappearing dry matter for samples that have been ensiled and from a cannulated cow. The authors again found a strong correlation between the predicted CSPS and measured dry matter disappearance.

There are a number of works for stover particles and common for them is that with computer vision they show that mechanical sieving underestimate the true length of the stover particles. In [16, 17], samples of corn and grass silage harvested at three different TLOCs, were analysed with the MATLAB image processing toolbox after being sorted using a separator and spread out on a flat surface. In addition to showing the underestimation, [17] also found that combining sorting and the image processing techniques improved the mass and size estimation of the stover samples. In [18] the authors investigated three longer TLOCs with image processing after sorting and spreading



## 2. State-of-the-art

out particles. The particles were found using Normalised Multiscale Bending Energy which provides features for the contour morphology from which the shape and size were extracted.

In Table 2.1 we summarise the above state-of-the-art in terms of the steps required, time required for measurements, where the measurement can be performed and the final metric. A number of the works, for example in [13], have additional steps when proving their methodology, such as freezing and thawing samples, however, we only include those that would be required by a farmer when conducting the measurement in a real-world scenario. From the table it can be seen that in the best case a quality estimate for a sample can be acquired in 1-2 hours. However, for most works [16–18] before the image analysis can be performed, likely samples would need to be transported off-site as machinery for mechanical sieving is generally heavy and cannot be moved. Whereas, [13] does potentially allow for CSPS estimation in the field through a smartphone application. It does however have a number of manual steps and requires separated kernel fragments to be placed on a black surface. Furthermore, thresholding of kernel fragments on the black surface can be sensitive to variations in lighting conditions if the analysis is performed in the field.

**Table 2.1:** Overview of the computer vision works for monitoring corn silage quality and their key characteristics.

Work	Steps Required	Time Required (hours)	Site	Metrics
[13]	Hydrodynamic separation Physical separation of fragments Image capture and analysis	1-2	In-field (requires black surface)	CSPS
[15]	Hydrodynamic separation Physical separation of fragments Image capture and analysis +Potential Ensiling +Potential Rumination	1-2 +days for ensiling +hours for rumination	In-field (requires black surface) +Silo	CSPS
[16]	Mechanical sieving Image capture and analysis	1-2 +potential transport	Laboratory Off-site	Mean particle length
[17]	Mechanical sieving Image capture and analysis	1-2 +potential transport	Laboratory Off-site	Mean particle length Comb. with sieving
[18]	Mechanical sieving Image capture and analysis	1-2 +potential transport	Laboratory Off-site	Mean particle length

As can be seen in the table current solutions are far from automatic. Moreover, they all require significant amount of time to perform, which reduce their usefulness. This directly motivates our work where we aim at an automatic system with fast processing.

## 2.2 Other Examples of Harvest Quality

Looking into the broader domain of not only WPCS and also using other sensors, there are a number of works that aim to measure quality of silage from a forage harvester.

Hyperspectral imaging is widely used in agriculture for a number of applications, including quality and yield measurement, and allows for capture of up to hundreds of bands that can exceed the visible spectrum [19]. For corn plants, the WPCS dry matter content and other quality aspects are measured with near-infrared spectroscopy (NIRS) directly on the machine in [20]. In [21] NIRS is used to adjust the TLOC during harvesting in order to provide better packing when stored in silos. The dry matter yield and crop quality have been measured with hyperspectral sensors off-nadir at multiple heights and angles in [22]. The Normalised Difference Vegetation Index together with plant height have been used to estimate the dry matter yield at an early stage of plant growth [23]. Looking into other examples of harvested silage it can be seen that hyperspectral imaging has been used to measure dry matter and other characteristics in other crops such as grass [24–29] and alfalfa [30, 31]. Other sensors have also been used for measuring silage such as X-ray [32], microwaves [33], mechanical displacement sensors [34, 35] and flow sensors [36]. While sensors such as NIRS can provide a quality measurement and additional nutrient information which is often not possible with an RGB camera, they are generally more expensive and can have a limited image resolution [37].

There are a number of works that adopt computer vision in other harvest quality applications. In relation to corn plants, [38] uses colour and shape features to train a maximum likelihood estimator to segment and classify between normal and damaged corn. However similar to works with WPCS, samples must be separated and spread out on a flat surface before analysis can be conducted. In [39] the level of corn kernel losses spread onto the field from a combine harvester was determined with a Faster R-CNN object detector [40] with a ResNet50 [41] backbone. An approach to monitor the digestive health of dairy cows in faecal samples was done in [42] by classifying fibre and corn content with deep learning and transfer learning. These two deep learning methods are successful in less controlled environments, for example in [39] the camera is mounted on the back of the harvester capturing images of the ground and in [42] corn particles are not required to be extracted from the faecal matter.

For the quality of rice and grains there are examples of both hand-crafted features and machine learning, however, all of them are conducted in a laboratory setup requiring manual sample preparation steps before images are captured. For example, the PSD of a number of biomasses, including rice, was found on samples spread out on a flat-bed scanner from which Feret's diameter could be calculated [43]. Rice grades have been classified using morphological features to train a Support Vector Machine (SVM) in [44, 45] and segmented with colour and shape features in [46]. Classification of different types of grains mixed in images has been investigated with a flat-bed scanner together with size, colour and brightness features [47]. The grades

have also been determined by training artificial neural networks with colour and texture [48], size, colour and shape [49], and colour and morphological features [50].

In agriculture there are numerous examples of adopting modern deep learning in applications similar to harvesting. Many of them exhibit strong results despite being present in challenging environments with clutter and occlusion, for example, from surrounding leaves. For example, [51] estimated the number of corn kernels on an entire ear of corn before harvesting with a sliding window Convolutional Neural Network (CNN). The biomass and crop composition is estimated with a modified VGG-16 for semantic segmentation in [52]. In [53] the quality of small grains was classified with an ensemble of networks. Classification with CNNs are used to identify different plant species [54] and conduct plant phenotyping [55]. Fine-grain classification of leaves by combining hand-crafted features and CNNs was conducted in [56]. Finally, there has been a number of examples of using deep learning in remote sensing data for estimating crop yields, including CNNs trained on NIRS for rice yields and for other grain parameters such as moisture and protein [57, 58]. For citrus fruits, colour and contour features were extracted to detect the fruits still on the trees in [59]. An adapted YOLOv4 [60] provided improved results, especially on smaller fruits, when compared to YOLOv3 [61] and Faster R-CNN [40] in [62]. In [63] a Faster R-CNN [40] with ResNet101 [41] detected highly occluded tomatoes on the plants, in [64] a modified YOLOv3 [61] produced circular bounding-boxes around the fruit and [65] used a modified Inception-ResNet [66] to count the number of tomatoes in images after training only on synthetic images. The detection of sweet peppers was conducted with two finetuned Faster R-CNNs [40] with VGG [67] backbones on RGB and NIRS images respectively in [68]. A Faster R-CNN [40] has also been used for detecting sweet peppers in [69] with an additional layer estimating the ripeness of the fruit. A number of variants of the YOLO detector [70], covering trade-offs in accuracy and speed, detected musk melons in [71]. In [72] a Faster R-CNN [40] detected passion fruits at multiple scales allowing for improvements on small fruits. Segmentation of apples for yield estimation was conducted with a custom CNN architecture in [73] and with a Faster R-CNN [40] in [74]. Mango detection and yield estimation was done with a Faster R-CNN [40] in [75], the authors also used a LiDAR to determine masks for trees in the canopy to which the detections could be matched. In [76] grape clusters were segmented with a Mask R-CNN [77] and tracked in video recordings.

Finally, there are a number of examples in the industry of leading agricultural manufacturers developing quality sensors on harvesters. A NIRS sensor is available on multiple forage harvesters to measure nutrient contents in the crop for Fendt [78], New Holland [79], CLAAS [80] and John Deere [81]. However, no options exist for estimating the PSD or metrics such as CSPS or

peNDF in the field. On the combine harvester there are multiple cameras that can measure the potential losses from broken grains and unwanted non-grain materials, such as from Fendt [82], CLAAS [83] and John Deere [84].

It is clear that within agriculture there is considerable activity with various sensors and computer vision. However, so far none have solved the problem of automatic estimation of corn silage quality, and hence this is the focus of our work.

### 2.3 Particle Size Distribution in other Domains

Motivated by the desire to use object recognition methods for efficiently monitoring WPCS quality we investigate applications in other domains where objects exist in cluttered scenes.

Firstly, an industry with numerous works covering quality is material aggregates. In [85, 86] the authors estimated the PSD of iron ore transported on a conveyor belt based on shape and size features. A number of hand-crafted features are extracted to train an SVM for estimation of iron ores on a conveyor in [87]. Iron ore pellets have also been segmented with a custom lightweight U-Net [88] to estimate the size distribution in [89]. Next, in medical images relevant applications include segmentation of brains and brain tumors in MRI images with a custom CNN [90, 91]. Finally, for crowd-counting deep learning has also shown impressive results such as in [92] that adopt a Feature Pyramid Network [93] and in [94] where faces are counted with a custom CNN.

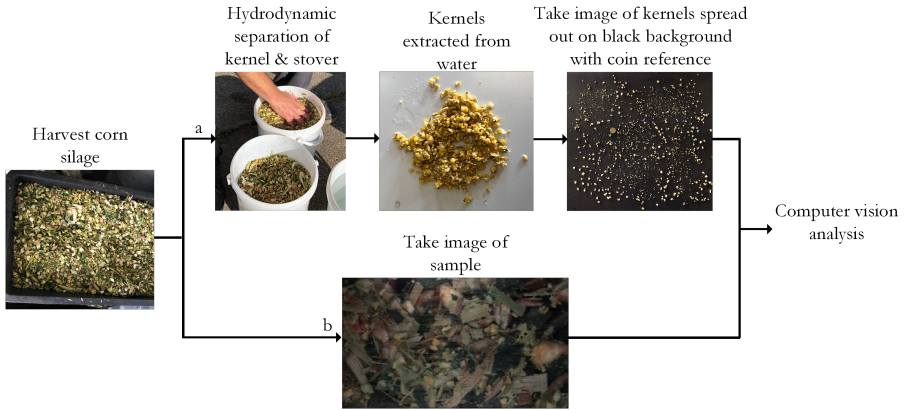
There are a number of examples of object recognition in crowded scenes over the previous sections, however, determining accurate size characteristics can be difficult due to potential sources of error that can appear. For example, particle sizes can be underestimated when they are occluded by other instances, where in [86] this was addressed by determining features that matched covered instances, such as a longer aspect ratio, and filtering predictions.

## 3 Contributions

Based upon the motivation to investigate if harvest quality monitoring can be automated with deep learning on RGB images the current practices by farmers and trends in research have been covered. For WPCS, a farmer can measure the quality by using methods, such as sieving, to separate the kernel and stover fragments. However, this often requires cumbersome and time-consuming sample preparation steps or sending samples to an off-site laboratory. Research does exist, as covered in Section 2.1, that aim to determine the quality of either kernel fragmentation or chopped stover with computer

### 3. Contributions

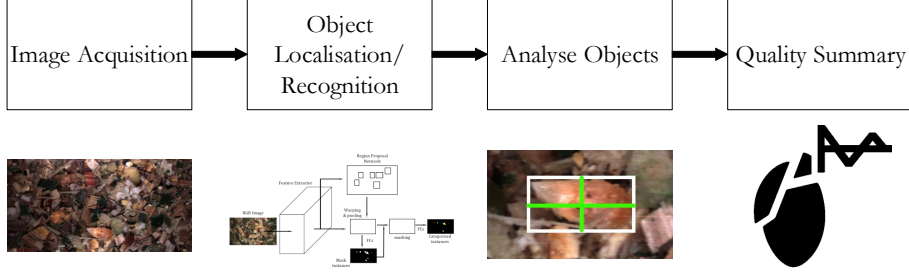
vision, however, all require the two portions of the silage to be separated from each other and spread out on a surface before capturing images. To speed up this process and place less requirements on manual steps, we are inspired to investigate if the quality can be measured in images of non-separated WPCS. Additionally, due to recent advances and promising results in similar applications, we aim to utilise deep learning to localise relevant objects with CNNs. In Figure 2.4 the differences between previous methodologies and our approach is highlighted. Firstly, Figure 2.4(a) shows the current steps required for measuring kernel fragmentation with the computer vision approach presented in [13, 14], here, kernels and stover are separated using hydrodynamic separation, then kernels are extracted from the water and the fragments are spread out on a black background before an image is captured for analysis. Instead, we propose to take an image directly of a WPCS sample, as visualised in Figure 2.4(b), followed by computer vision analysis localising objects relevant to either kernel or stover quality. The removal of manual steps for sample preparation can lead to a system for estimating the corn silage quality quickly in the field. However, manual separation does have some benefits, such as each particle can be measured accurately without occluding each other. Therefore, in this PhD steps were taken in the data collection, model development and evaluation stages to create a robust and well-performing system despite the challenging cluttered scenes.



**Fig. 2.4:** The differences in sample preparation between previous state-of-the-art and our proposed approach. (a) Previously, as in [13], kernels and stover are first separated using the hydrodynamic method, kernel fragments are removed from the water and spread out on a black background with a coin for size reference before images are captured for analysis. In (b) our proposed method takes an image directly of the sample and requires no sample preparation.

Without the need for sample separation we propose a system following the generic industrial computer vision definition presented in Section I.2.1 in Figure 2.5. In such a system an image is acquired of a WPCS sample di-

rectly from the harvester, object recognition through a deep learning network localises and classifies kernel or stover particles, the predicted particles are analysed inspired by industry standards and finally a summary of the quality of the WPCS samples is estimated.



**Fig. 2.5:** Overview of the computer vision system proposed in this PhD for monitoring WPCS quality. Figure adapted with images from [1, 95] and © CLAAS.

Our first contribution is to introduce the first work on measuring the processing of kernel fragments in images of non-separated WPCS samples. As covered, previous approaches such as [13, 14] require a number of time-consuming steps for separating and spreading of kernels. We instead evaluate two forms of object recognition networks for the task on our images, namely, a Region-based Fully Convolutional Network (R-FCN) [96] with an ResNet101 [41] backbone and a Multi-task Network Cascade (MNC) [97] with AlexNet [98] in the form of bounding-box and instance segmentation networks respectively. In Figure 2.6 we show example predictions from the two networks where white outlines indicate ground truth annotations. The methods show promising results in terms of precision and recall, and allow us to extract object instance characteristics such that we can estimate the CSPS over a sample of images. We show a first indication of strong correlation between model CSPS against CSPS estimated from annotations. Further details can be seen in Paper A.

We also propose to improve the use of two-stage recognition networks for measuring the quality of WPCS in our non-separated samples. Again, the use of these networks allow us to extract characteristics about size and shape for individual predicted instances. A requirement that is necessary if a PSD should be captured allowing for CSPS estimation. For both kernel fragmentation and stover overlengths, we found considerable improvement in terms of Average Precision (AP) measured from annotations and correlation analysis against physically sieved measurements. The improvements came through investigations in training numerous Faster R-CNNs [40] with an Inceptionv2 [99] backbone with strategies in data separation, transfer learning and anchor tuning in the Region Proposal Network (RPN) [40]. However,

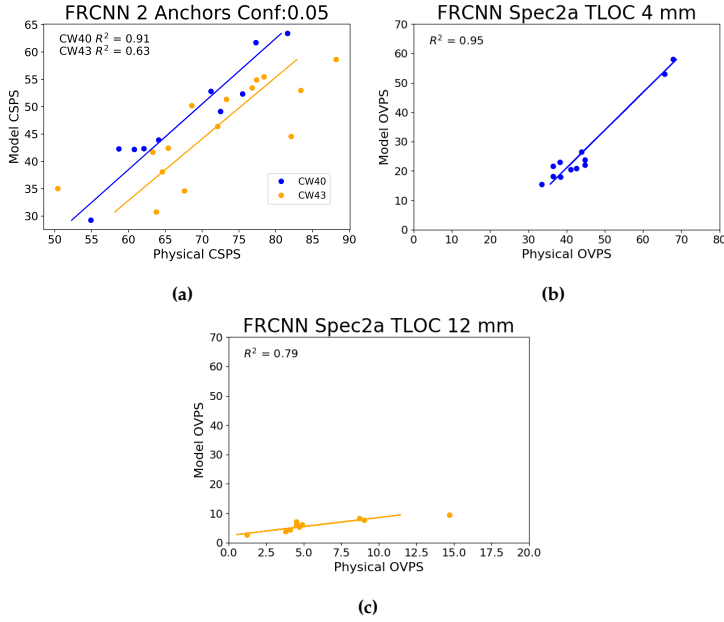
### 3. Contributions



**Fig. 2.6:** Example predictions for models trained for kernel fragmentation from a R-FCN bounding-box detector (a) and MNC instance segmentation network (b). Images from [1].

our improvements were not consistent in all cases between annotation and physical evaluation, indicating the importance of evaluating the system independent of annotations. With trained networks for the tasks, we estimate the CSPS for kernel fragmentation and a measure we introduced, the Overlength Particle Score (OVPS), for stover overlengths. OVPS differs from CSPS in that the aim is not to detect all particles. Stover particles account for the vast majority of pixels in our images, therefore, full PSD was deemed to be infeasible for this task. Therefore, we train models to predict the level of overlengths, or how many stover particles are too long. Additionally, we were motivated by a potentially varying TLOC strategy for promoting peNDF from a given farmer depending on their specific crop and farm structure. Therefore, we evaluated stover OVPS based on a variable definition of overlengths given the TLOC at the time of harvest. Through close collaboration with machine experts at the Industrial PhD host company, a good compromise definition of  $1.5 \times \text{TLOC}$  was made. For both CSPS and OVPS, we show a strong correlation against sieved samples harvested at a number of machine settings. For kernel fragmentation, compared to a naive standard training approach, we improved AP by 11.3% and  $r^2$  to physical CSPS by 26.6% to a strong correlation of 0.66. Additionally for stover overlengths, with our model improvements, the AP was improved by up to 45.2% and correlation to physical samples by 132.4% to an  $r^2$  of 0.95 and 0.79 at TLOC 4 mm and 12 mm respectively. Figure 2.7 shows the correlation analysis for both CSPS and OVPS. The CSPS analysis is shown for two harvest weeks in Figure 2.7(a), while OVPS is shown for TLOC 4 mm in 2.7(b) and TLOC 12 mm in 2.7(c). The work is covered in more detail in Paper B.

Additionally, in Paper B we include an appendix covering further design choices for two-stage object recognition networks. Kernel fragments are classified in our networks based upon an axis length mimicking the sieving methods. We therefore evaluate differences based upon classifying with the major, minor and mean axis of a predicted object. Here, we determine



**Fig. 2.7:** Correlation analysis between the Faster R-CNN models and physical measurements for CSPS (a), OVPS at TLOC 4 mm (b) and TLOC 12 mm(c). Images from [2].

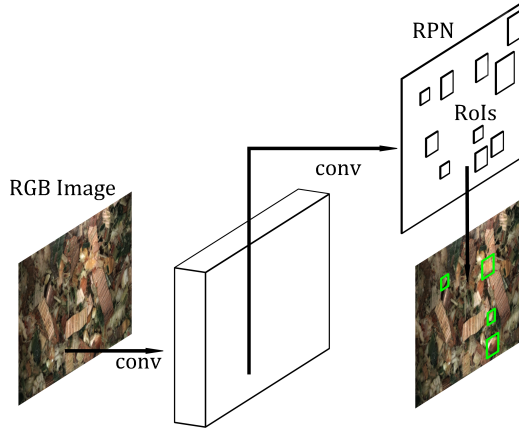
optimal results for multiple network architectures based on major axis classification. Additionally, we investigate differences between Faster R-CNNs [40] with either an Inceptionv2 [99] or ResNet50 [41] backbone, with better models being found with the former architecture. Furthermore, we evaluated the differences between CSPS and OVPS for our Faster R-CNN [40] bounding-box models and for Mask R-CNN [77] segmentation mask producing models. Here, higher correlation was found with bounding-box models despite the possibility of overestimating particles in comparison to the finer-localised segmentation masks. Lastly, the effect of lowering the image resolution in regards to speed and performance was presented showing significant decreases in AP and correlation.

Our works so far have concentrated on two-stage networks. These have produced promising results but some aspects of the networks can be redundant if the aim is to estimate quality based on sieving standards. The networks have been trained to perform both classification and fine-grain localisation. The accurate localisation in the form of bounding-boxes or masks have allowed us to estimate the harvest quality through metrics such as CSPS. However, sieving does not aim to find the precise size of each particle but rather classify which sieving pan a particle would end in. Therefore, we developed a novel network, which we have named SieveNet, that can effi-



### 3. Contributions

ciently and accurately classify a kernel fragment into a pre-defined number of sieves without performing fine-grain localisation. Figure 2.8 visualises the differences between two-stage recognition and our approach. We remove the modules for performing mask regression and classification and instead classify directly from a modified RPN into sieve classes.



**Fig. 2.8:** Overview of our architecture for directly classifying sieve sizes for kernel fragments without the need for box/mask regression or classification modules. Image adapted from [1].

This is realised by presenting a novel matching algorithm allowing for sieve-based anchor matching during training. Then during inference, particles are classified directly into a sieve class without bounding-box or mask regression. The approach shows an improvement in inference timings and a strong correlation when estimating CSPS. Further details can be seen in Paper C.

From these works our main scientific contributions within monitoring corn silage quality can be summarised as:

- The first works on estimating the quality of harvested corn silage in RGB images without the need for separating kernel and stover particles.
  - First presented in Paper A: Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Images.
  - Algorithm improvements in Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
  - Further improvements in Paper C: SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage.

## References

- We propose robust two-stage networks for the task of localising kernel fragments and stover overlengths across harvest seasons and machine settings.
  - Kernel fragments evaluated in Paper A: Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Image.
  - Kernel fragments and stover overlengths localised in Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
- We show significant improvements to kernel and stover overlength quality monitoring in two-stage networks by investigating strategies for data separation and transfer learning, together with tuning parameters in the Region Proposal Network with respective shape and size characteristics for the two tasks.
  - Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
- We present a novel sieve-based matching algorithm allowing us to train models for efficient estimation of kernel fragmentation quality.
  - Paper C: SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage.

## References

- [1] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [2] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Anchor tuning in faster r-cnn for measuring corn silage physical characteristics," *Computers and Electronics in Agriculture*, vol. 188, p. 106344, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169921003616>
- [3] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Sievenet: Estimating the particle size distribution of kernel fragments in whole plant corn silage," *Accepted/in press at the 17th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022.
- [4] P. Savoie, K. Shinnars, and B. Binversie, "Hydrodynamic separation of grain and stover components in corn silage," *Appl Biochem Biotechnol.*, vol. 113-116, pp. 41–54, 2004.

## References

- [5] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," *In: Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [6] D. Mertens, "Creating a system for meeting the fiber requirements of dairy cows," *Journal of Dairy Science*, vol. 80, no. 7, pp. 1463–1481, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030297760752>
- [7] G. Dias Junior, L. Ferraretto, G. Salvati, L. de Resende, P. Hoffman, M. Pereira, and R. Shaver, "Relationship between processing score and kernel-fraction particle size in whole-plant corn silage," *Journal of Dairy Science*, vol. 99, no. 4, pp. 2719–2729, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002203021600120X>
- [8] L. Ferraretto, R. Shaver, and B. Luck, "Silage review: Recent advances and future technologies for whole-plant and fractionated corn silage harvesting," *Journal of Dairy Science*, vol. 101, no. 5, pp. 3937–3951, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030218303199>
- [9] B. Lammers, D. Buckmaster, and A. Heinrichs, "A simple method for the analysis of particle sizes of forage and total mixed rations," *Journal of Dairy Science*, vol. 79, no. 5, pp. 922–928, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030296764421>
- [10] Penn State Extension, "Penn State Particle Separator," <https://extension.psu.edu/penn-state-particle-separator> (accessed February 10, 2020), 2016.
- [11] D. Maulfair and A. Heinrichs, "Review: Methods to measure forage and diet particle size in the dairy cow," *The Professional Animal Scientist*, vol. 28, no. 5, pp. 489–493, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S108074461530396X>
- [12] ASABE, "Method of determining and expressing particle size of chopped forage materials by screening," *ANSI/ASAE*, vol. S424.1, p. 663–665.
- [13] J. L. Drewry, B. D. Luck, R. M. Willett, E. M. Rocha, and J. D. Harmon, "Predicting kernel processing score of harvested and processed corn silage via image processing techniques," *Computers and Electronics in Agriculture*, vol. 160, pp. 144 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918311955>

## References

- [14] B. Luck, "Silagesnap." [Online]. Available: <https://https://wimachineryextension.bse.wisc.edu/precision-agriculture/silagesnap/> (accessed on 9 March 2021)
- [15] B. D. Luck, J. L. Drewry, R. D. Shaver, R. M. Willett, and L. F. Ferraretto, "Predicting in situ dry matter disappearance of chopped and processed corn kernels using image-analysis techniques," *Applied Animal Science*, vol. 36, no. 4, pp. 480–488, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590286520300847>
- [16] P. Savoie, M.-A. Audy-Dubé, G. Pilon, and R. Morissette, "Chopped forage particle size analysis in one, two and three dimensions," 01 2013.
- [17] P. Savoie, M. Audy-Dubé, G. Pilon, and R. Morissette, "Length distribution and other dimensional parameters of chopped forage by image analysis," *Transactions of the ASABE*, vol. 57, no. 6, pp. 1549–1555, 2014.
- [18] M. Audy, P. Savoie, F. Thibodeau, and R. Morissette, "Size and shape of forage particles by image analysis and normalized multiscale bending energy method," *American Society of Agricultural and Biological Engineers Annual International Meeting 2014, ASABE 2014*, vol. 2, pp. 820–830, 01 2014.
- [19] K. L. M. Ang and J. K. P. Seng, "Big data and machine learning with hyperspectral information in agriculture," *IEEE Access*, vol. 9, pp. 36 699–36 718, 2021.
- [20] R. Welle, W. Greten, B. Rietmann, S. Alley, G. Sinnaeve, and P. Dardenne, "Near-infrared spectroscopy on chopper to measure maize forage quality parameters online," *Crop Science*, vol. 43, no. 4, pp. 1407–1413, 2003. [Online]. Available: <https://access.onlinelibrary.wiley.com/doi/abs/10.2135/cropsci2003.1407>
- [21] M. Digman and K. Shinnars, "Real-time moisture measurement on a forage harvester using near infrared reflectance spectroscopy," *Transactions of the ASABE*, vol. 51, pp. 1801–1810, 2008.
- [22] D. Perbandt, T. Fricke, and M. Wachendorf, "Off-nadir hyperspectral measurements in maize to predict dry matter yield, protein content and metabolisable energy in total biomass," *Precision Agriculture*, vol. 12, pp. 249–265, 2011.
- [23] M. Islam and S. Garcia, "Prediction of dry matter yield of hybrid forage corn grown for silage," *Crop Science*, vol. 54, no. 5, pp. 2362–2372, 2014. [Online]. Available: <https://access.onlinelibrary.wiley.com/doi/abs/10.2135/cropsci2013.10.0710>

## References

- [24] D. Perbandt, T. Fricke, and M. Wachendorf, "Development and validation of near-infrared spectroscopy for the prediction of forage quality parameters in *lolium multiflorum*," *PeerJ*, vol. 5, no. e3867, 2017.
- [25] C. Smith, S. Karunaratne, P. Badenhorst, N. Cogan, G. Spangenberg, and K. Smith, "Machine learning algorithms to predict forage nutritive value of in situ perennial ryegrass plants using hyperspectral canopy reflectance data," *Remote Sensing*, vol. 12, no. 6, 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/6/928>
- [26] N. Viljanen, E. Honkavaara, R. Nasi, T. Hakala, O. Niemelainen, and J. Kaivosoja, "A novel machine learning method for estimating biomass of grass swards using a photogrammetric canopy height model, images and vegetation indices captured by a drone," *Agriculture*, vol. 8, no. 5, 2018. [Online]. Available: <https://www.mdpi.com/2077-0472/8/5/70>
- [27] H. E. Johnson, D. Broadhurst, D. B. Kell, M. K. Theodorou, R. J. Merry, and G. W. Griffith, "High-throughput metabolic fingerprinting of legume silage fermentations via fourier transform infrared spectroscopy and chemometrics," *Applied and Environmental Microbiology*, vol. 70, no. 3, pp. 1583–1592, 2004. [Online]. Available: <https://aem.asm.org/content/70/3/1583>
- [28] R. Oliveira, R. Nasi, O. Niemelainen, L. Nyholm, K. Alhonoja, J. Kaivosoja, N. Viljanen, T. Hakala, S. Nezami, L. Markelin, L. Jauhiainen, and E. Honkavaara, "Assessment of rgb and hyperspectral uav remote sensing for grass quantity and quality estimation," *Remote Sensing and Spatial Information Sciences*, vol. 42, no. 2, pp. 489–494, 2019.
- [29] G. Togeiro de Alckmin, L. Kooistra, R. Rawnsley, S. de Bruin, and A. Lucieer, "Retrieval of hyperspectral information from multispectral data for perennial ryegrass biomass estimation," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/24/7192>
- [30] C. Keith, K. Repasky, R. Lawrence, S. Jay, and J. Carlsten, "Monitoring effects of a controlled subsurface carbon dioxide release on vegetation using a hyperspectral imager," *International Journal of Greenhouse Gas Control*, vol. 3, no. 5, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1750583609000279>
- [31] L. Feng, Z. Zhang, Y. Ma, Q. Du, P. Williams, J. Drewry, and B. Luck, "Alfalfa yield prediction using uav-based hyperspectral imagery and ensemble learning," *Remote Sensing*, vol. 12, no. 12, 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/12/2028>

## References

- [32] K. J. Wild, T. Schmiedel, S. Lautenschlager, M. Stein, and J. K. Schueller, "X-ray technique for simultaneous mass flow measurements and foreign body detection in a self-propelled forage chopper," in *2015 ASABE Annual International Meeting*. American Society of Agricultural and Biological Engineers, 2015, p. 1.
- [33] M. Matsuo, A. Osada, and S. Kon, "Non-destructive prediction of forage crop moisture contents using microwave transmitted signals with a microstrip transmission line sensor," *Grassland Science*, vol. 66, no. 4, pp. 225–230, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/grs.12271>
- [34] H. Martel, P. Savoie *et al.*, "Sensors to measure mass-flow-rate through a forage harvester," *Canadian Agricultural Engineering*, vol. 42, no. 3, pp. 123–130, 2000.
- [35] D. Ehlert, "Pa—precision agriculture: Advanced throughput measurement in forage harvesters," *Biosystems Engineering*, vol. 83, no. 1, pp. 47–53, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511002901017>
- [36] E. Long, Q. Ketterings, D. Russell, F. Vermeylen, and S. DeGloria, "Assessment of yield monitoring equipment for dry matter and yield of corn silage and alfalfa/grass," *Precision Agriculture*, vol. 17, pp. 546–563, 2016.
- [37] H. G. Yakubu, Z. Kovacs, T. Toth, and G. Bazar, "The recent advances of near-infrared spectroscopy in dairy production—a review," *Critical Reviews in Food Science and Nutrition*, vol. 0, no. 0, pp. 1–22, 2020, pMID: 33043681. [Online]. Available: <https://doi.org/10.1080/10408398.2020.1829540>
- [38] X. Li, B. Dai, H. Sun, and W. Li, "Corn classification system based on computer vision," *Symmetry*, vol. 11, no. 4, 2019. [Online]. Available: <https://www.mdpi.com/2073-8994/11/4/591>
- [39] N. S. Monhollen, K. J. Shinnors, J. C. Friede, E. M. Rocha, and B. D. Luck, "In-field machine vision system for identifying corn kernel losses," *Computers and Electronics in Agriculture*, vol. 174, p. 105496, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920302702>
- [40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

## References

- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
- [42] G. A. Atkinson, L. N. Smith, M. L. Smith, C. K. Reynolds, D. J. Humphries, J. M. Moorby, D. K. Leemans, and A. H. Kingston-Smith, "A computer vision approach to improving cattle digestive health by the monitoring of faecal samples," *Scientific reports*, vol. 10, no. 1, p. 17557, 2020.
- [43] C. Igathinathane, L. Pordesimo, E. Columbus, W. Batchelor, and S. Sokhansanj, "Sieveless particle size distribution analysis of particulate materials through computer vision," *Computers and Electronics in Agriculture - COMPUT ELECTRON AGRIC*, vol. 66, pp. 147–158, 05 2009.
- [44] H. Kaur and B. Singh, "Classification and grading rice using multi-class svm," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, pp. 1–5, 2013.
- [45] F. Antonucci, S. Figorilli, C. Costa, F. Pallottino, A. Spanu, and P. Mene-satti, "An open source conveyor belt prototype for image analysis-based rice yield determination," *Food and Bioprocess Technology*, vol. 10, pp. 1–8, 02 2017.
- [46] P. Dubosclard, S. Larnier, H. Konik, A. Herbulot, and M. Devy, "Automatic visual grading of grain products by machine vision," *Journal of Electronic Imaging*, vol. 24, p. 061116, 11 2015.
- [47] G. Dalen, "Determination of the size distribution and percentage of broken kernels of rice using flatbed scanning and image analysis," *Food Research International*, vol. 37, pp. 51–58, 06 2004.
- [48] N. Visen, J. Paliwal, D. Jayas, and N. White, "Image analysis of bulk grain samples using neural networks," *Canadian Biosystems Engineering / Le Genie des biosystemes au Canada*, vol. 46, 01 2003.
- [49] B. Anami and D. Savakar, "Effect of foreign bodies on recognition and classification of bulk food grains image samples," *Journal of Applied Computer Science & Mathematics*, vol. 3, 01 2009.
- [50] C. Lee, L. Yan, T. Wang, S. Lee, and C. Park, "Intelligent classification methods of grain kernels using computer vision analysis," *Measurement Science and Technology*, vol. 22, p. 064006, 05 2011.
- [51] S. Khaki, H. Pham, Y. Han, A. Kuhl, W. Kent, and L. Wang, "Convolutional neural networks for image-based corn kernel detection and counting," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/9/2721>

## References

- [52] A. Mortensen, M. Dyrmann, H. Karstoft, R. Jørgensen, and R. Gislum, "Semantic segmentation of mixed crops using deepconvolutional neural network," in *InProc. of the International Conf. of Agricultural Engineering*. CIGR, 2017.
- [53] Y. Zhou, W. Wu, J. Zou, J. Qiao, and J. Cheng, "Weighted ensemble networks for multiview based tiny object quality assessment," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 6, p. e5995, 2021, e5995 CPE-20-0377.R1. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5995>
- [54] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 452–456.
- [55] M. P. Pound, J. A. Atkinson, A. J. Townsend, M. H. Wilson, M. Griffiths, A. S. Jackson, A. Bulat, G. Tzimiropoulos, D. M. Wells, E. H. Murchie, T. P. Pridmore, and A. P. French, "Deep machine learning provides state-of-the-art performance in image-based plant phenotyping," *GigaScience*, vol. 6, no. 10, pp. 1–10, 2017.
- [56] D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 797–804.
- [57] S. Assadzadeh, C. Walker, L. McDonald, P. Maharjan, and J. Panozzo, "Multi-task deep learning of near infrared spectra for improved grain quality trait predictions," *J. Near Infrared Spectrosc.*, vol. 28, no. 5, pp. 275–286, Oct 2020. [Online]. Available: <http://www.osapublishing.org/jnirs/>
- [58] Q. Yang, L. Shi, J. Han, Y. Zha, and P. Zhu, "Deep convolutional neural networks for rice grain yield estimation at the ripening stage using uav-based remotely sensed images," *Field Crops Research*, vol. 235, pp. 142–153, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S037842901831390X>
- [59] J. Lu and N. Sang, "Detecting citrus fruits and occlusion recovery under natural illumination conditions," *Computers and Electronics in Agriculture*, vol. 110, pp. 121–130, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169914002774>
- [60] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.



- [61] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.
- [62] W. Chen, S. Lu, B. Liu, G. Li, and T. Qian, "Detecting citrus fruits and occlusion recovery under natural illumination conditions," *Scientific Programming*, vol. 2020, 2020.
- [63] Y. Mu, T.-S. Chen, S. Ninomiya, and W. Guo, "Intact detection of highly occluded immature tomatoes on plants using deep learning techniques," *Sensors*, vol. 20, no. 10, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/10/2984>
- [64] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "Yolo-tomato: A robust algorithm for tomato detection based on yolov3," *Sensors*, vol. 20, no. 7, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/7/2145>
- [65] M. Rahnemounfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors (Basel, Switzerland)*, vol. 17, 04 2017.
- [66] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.
- [67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [68] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. Mccool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, p. 1222, 08 2016.
- [69] M. Halstead, C. McCool, S. Denman, T. Perez, and C. Fookes, "Fruit quantity and ripeness estimation using a robotic vision system," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2995–3002, 2018.
- [70] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.
- [71] O. M. Lawal, "Yolomuskmelon: Quest for fruit detection speed and accuracy using deep learning," *IEEE Access*, vol. 9, pp. 15 221–15 227, 2021.
- [72] S. Tu, J. Pang, H. Liu, Y. Zhuang, N. Chen, C. Zheng, H. Wan, and Y. Xue, "Passion fruit detection and counting based on multiple scale faster r-cnn using rgb-d images," *Precision Agriculture*, vol. 21, pp. 1072–1091, 2020.

## References

- [73] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1039–1060, 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21699>
- [74] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," 05 2017, pp. 3626–3633.
- [75] M. Stein, S. Bargoti, and J. Underwood, "Image based mango fruit detection, localisation and yield estimation using multiple view geometry," *Sensors*, vol. 16, no. 11, 2016. [Online]. Available: <https://www.mdpi.com/1424-8220/16/11/1915>
- [76] T. Santos, L. Souza, A. Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, 2020.
- [77] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.
- [78] Fendt, "Fendt katana," accessed on March 12 2021. [Online]. Available: <https://www.fendt.com/int/geneva-assets/article/95936/65901-fendtkatana-1901-en-v2.pdf>
- [79] N. Holland, "Fr forage cruiser," accessed on March 12 2021. [Online]. Available: <https://assets.cnhindustrial.com/nhag/eu/en-uk/assets/pdf/forage-harvesters/fr-forage-cruiser-brochure-uk-en.pdf>
- [80] CLAAS, "Jaguar 990-930 data management," accessed on March 12 2021. [Online]. Available: <https://www.claas.co.uk/products/forage-harvesters/jaguar990-930/datamanagemet#cid2076186>
- [81] J. Deere, "John deere harvestlab 300," accessed on March 12 2021. [Online]. Available: <https://www.deere.com/en/technology-products/precision-ag-technology/data-management/harvest-lab-constituent-sensing/>
- [82] Fendt, "Fendt ideal," accessed on March 12 2021. [Online]. Available: <https://www.fendt.com/us/combindes/ideal-cab>
- [83] CLAAS, "Lexion," accessed on March 12 2021. [Online]. Available: <https://www.claas.co.uk/blueprint/servlet/blob/2392078/b5287570e9da4a3dc72ea3e142aeacf3/405581-23-dataRaw.pdf>
- [84] J. Deere, "Grain quality guaranteed," accessed on March 12 2021. [Online]. Available: <https://www.deere.com/international/en/campaigns/ag-turf/grain-quality-guaranteed/>

## References

- [85] T. Andersson, M. J. Thurley, and O. Marklund, "Visibility classification of pellets in piles for sizing without overlapped particle error," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 2007, pp. 508–514.
- [86] T. Andersson, M. J. Thurley, and J. E. Carlson, "A machine vision system for estimation of size distributions by weight of limestone particles," *Minerals Engineering*, vol. 25, no. 1, pp. 38–46, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0892687511003682>
- [87] A. Patel, S. Chatterjee, and A. Gorai, "Development of machine vision-based ore classification model using support vector machine (svm) algorithm," *Arabian Journal of Geosciences*, vol. 10, no. 107, 2017.
- [88] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.
- [89] J. Duan, X. Liu, X. Wu, and C. Mao, "Detection and segmentation of iron ore green pellets in images using lightweight u-net deep learning network," *Neural Computing and Applications*, vol. 32, pp. 5575–5790, 2020.
- [90] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.
- [91] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Išgum, "Automatic segmentation of mr brain images with a convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1252–1261, 2016.
- [92] T. W. Cenggoro, A. H. Aslamiah, and A. Yunanto, "Feature pyramid networks for crowd counting," *Procedia Computer Science*, vol. 157, pp. 175–182, 2019, the 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919310737>
- [93] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

## References

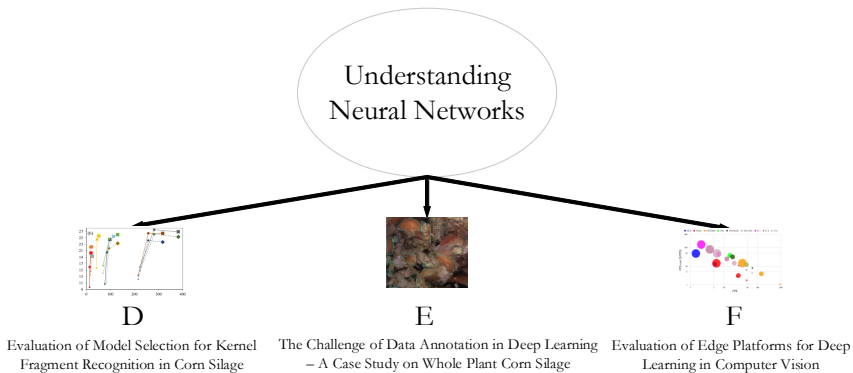
- [94] Y. Wang, J. Hou, X. Hou, and L. P. Chau, "A self-training approach for point-supervised object detection and counting in crowds," *IEEE Transactions on Image Processing*, vol. 30, pp. 2876–2887, 2021.
- [95] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "The challenge of data annotation in deep learning – a case study on whole plant corn silage," *Under review at MDPI Sensors*, 2021.
- [96] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," 12 2016, pp. 379–387.
- [97] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," 06 2016, pp. 3150–3158.
- [98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [99] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

## Chapter 3

# Understanding Neural Networks

### 1 Introduction

The aim of this part of the thesis is to understand a number of key aspects of computer vision systems adopting neural networks. The focus of the chapter is on the trade-off in state-of-the-art object recognition models, the challenge of building datasets for deep learning, and how to deploy models on an appropriate platform. As this work is part of an Industrial PhD, these research topics have been especially important in terms of deployability into the real-world. This chapter will give an introduction to the topics and an overview of the relevant state-of-the-art will be given. This will lead to covering our works, which are visualised in Figure 3.1, and present our contributions.



**Fig. 3.1:** Overview of the works covered in this chapter for understanding neural networks. Figure with images from [1–3].

When designing a computer vision system, care must be taken to reduce any unwanted variance in the system. If the captured images can be as consistent as possible it can lower requirements on model complexity as the networks can concentrate on learning the variations between objects. Variations specific to the image include the amount of illumination, viewpoint or quality and object variations can be the size and shape. State-of-the-art computer vision models are often developed and optimised towards benchmark challenges, such as ImageNet [4] and COCO [5], where objects are annotated in images collected from a large number of sources with variations in the capturing. However, industry models are typically deployed in scenarios where the company chooses the hardware related to their product and image variations are therefore decreased. Despite this, there can be differences between sensors in production and illumination in a given scenario. As models should perform optimally when deployed, care should be taken to address these variations to reduce any potential losses in accuracy, in addition to optimising towards object variation.

During my master's thesis I investigated creating an ensemble of R-FCN [6] networks that aimed to decrease individual ensemble member variance based on image quality and object size [7]. For image quality, two networks were trained for low and high quality for a number of distortions including Gaussian blur and JPEG compression. The distortions were estimated for an image with a no-reference Image Quality Assessment (IQA) approach using DeepIQA [8] and object sizes were estimated using sizes of proposals found with the RPN [9]. At inference time the DeepIQA and RPN were used to weight the output of individual ensemble members for a final prediction and led to increasing the AP in comparison to baseline non-ensemble approaches. This work laid the basis for my interest in working with real-world systems and adapting to the resulting variance, for example, the works presented in the previous chapter on monitoring WPCS through kernel and stover recognition. For additional details on work done in my master's, see the published paper in [10].

It is well established that deep learning models require large amounts of data to train and evaluate models. For a project to be successful, high quality and consistent annotations are key [11, 12]. Benchmark datasets such as ImageNet [13] and COCO [5] achieve this through processes which are highly-defined including multi-stage annotation and individuals with various roles. Naturally, this can be expensive, difficult to define and implement, especially in smaller projects with limited resources. Therefore, as covered earlier, due to the increasing popularity of adopting neural networks in computer vision systems, care should be placed on building and evaluating datasets.

Finally, there are trade-offs when developing models for deep learning systems in complexity, accuracy and speed. In addition, in real-world industrial scenarios it is important to take into account the deployability of the

model. There are a number of framework options for deep learning models for both training and inference such as TensorFlow and TensorFlow lite [14]. An optimal deep learning system in the real-world takes into account the performance of a model for the task at hand on a potential hardware platform.

## 2 State-of-the-art

### 2.1 Object Recognition with Neural Networks

To obtain an understanding of how to account for variational challenges in object recognition we start by giving an overview of the key recent works. Deep learning has been synonymous with computer vision in many applications since AlexNet won the ImageNet classification challenge in 2012 [15]. Within object recognition, the two-stage approach of first producing a number of object proposals followed by proposal classification and box refinement is well established. The most recent winners of the COCO [5] challenges follow this mantra, where [16] used Feature Pyramid Networks for bounding-box detection and [17] used a custom Mask R-CNN [18] named MegDetV2. The proposal module in a two-stage object recognition network is typically lighter weight as it allows for more complexity in other parts of the architecture, such as for classification or feature extraction. Due to increasing complexity in deep learning-based models this is necessary compared to a more traditional oversampling approach such as a sliding window. Proposal generation has been present in object recognition before deep learning, as covered in the comprehensive review in [19], including using superpixels in SelectiveSearch [20] and edge detection in EdgeBoxes [21]. However, since Faster R-CNN [9], the usage of the RPN has become the standard. In the RPN, a lightweight sliding window traverses the final feature map and at each sliding window location, anchor boxes with pre-defined shapes and sizes are regressed from which the class-agnostic probability of an object is computed. Afterwards, following non-maximum suppression, the top  $N$  boxes are further refined and classified into the classes defined during training, where typically  $N$  is the order of hundreds.

Open-source frameworks together with more powerful GPUs have been two of the reasons for deep learning success. Many frameworks have code and pre-trained models available that allow researchers to conduct fast prototyping and produce strong networks. Examples include the TensorFlow Object Detection API from Google [22] and Detectron2 from Facebook AI [23]. The frameworks have lower and higher complexity models, for example in TensorFlow, models that can be run on lower compute hardware are the Single Shot Multibox Detector (SSD) [24] or EfficientDet [25]. At the higher end, models such as Mask R-CNN [18] and ExtremeNet [26] are also available.

For most of the networks a number of feature extractors can be specified to further adjust the model complexity. Finding the appropriate model can be challenging when aiming to solve a specific task. Additionally, there are typically a number of requirements for a system in regards to accuracy and speed, but this is highly dependent on how challenging the task is and the quality of the data. The trade-offs in modern deep learning networks have been covered in [27], but the focus is on larger benchmark datasets and the findings may not be directly transferable to a specific task, such as WPCS quality. Other options exist, such as defining a custom architecture, but significant time can be spent determining parameters such as the number of layers or filter size. Alternatively, Neural Architecture Search has become increasingly popular where an algorithm defines the architecture of the network, however, work within computer vision has been largely related to classification and can have significant requirements on the training time [28, 29].

## 2.2 Building Datasets

As mentioned, benchmarks often have highly-defined processes to obtain a high quality annotated dataset. In Table 3.1 we provide an overview of some key processes and statistics for a number of benchmarks within object recognition. It can be seen that object classes vary between 20 to a few thousand, and annotated instances from tens of thousands to millions. Generally, the smaller datasets were less complex to collect. For example, the initial annotations for PASCAL VOC [30] were made by researchers at an in-person annotation event and the annotations in ADE20K [31] are annotated by a single person. However, we see that as datasets have grown in size so have the number of processes. Firstly, in ImageNet [13], the dataset for localisation set a basis for crowd-sourcing annotations through Amazon Mechanical Turk (AMT). Multiple roles were implemented including annotator and verifier. A four stage process was used for annotation, where first categories for objects were labelled at an image level. From this, annotators were asked to draw a single bounding-box on an image, which was checked for quality by a verifier. Additionally, a third verifier evaluated the coverage, if all instances were annotated in an image. Once both verifiers approve an image it is accepted into the dataset. In the COCO [5] instance segmentation dataset, a focus was placed on non-iconic images and a higher number of objects than previous benchmarks. This was also achieved with AMT and having roles for annotator and verifier. Category labelling was also used at an image level for the first step. Next, an annotator marked all instances in an image for one specific class. Then in the final stage, annotators were instructed to create a single mask per image from the markings. Multiple verifiers per mask was used before an annotation was accepted. This step also allowed the creators to remove specific annotators and their masks if quality was poor. In LVIS [32], a



dataset with focus on a large number of classes, a similar approach to COCO was taken. Annotators performed category labelling in a first step, but only marked a maximum of one instance at a time for an image. Next, all instances of a single class was marked, followed by instance segmentation for a single mask per image. Masks are then verified by multiple persons before acceptance. Finally, the final step used multiple verifiers to check for full object coverage. The Open Images dataset [33] contains both bounding-box and instance mask annotations. Their annotation process is only covered for boxes in [33], however, it is likely similar for masks. Category labelling is again used, but this is performed by a classifier model and verified by a human. In the next stage, an annotator is iteratively instructed to draw bounding-boxes for all instances of a single category. Lastly, another example of a large dataset is Objects365 [11]. Their first step is again category labelling which is followed by drawing boxes for all instances of a single class iteratively. A verifier is used to accept or reject all annotations. Additionally, a third role of examiner reviews the work periodically from annotators and verifiers. In addition, in Table 3.1 we see that multiple datasets incorporate training for the respective roles and inject gold standard sets to perform further verification. It is clear from Table 3.1 that current practices for developing large datasets is an extensive task. While they appear to gather annotations for a dataset of sufficient quality that have a benefit to the computer vision community, the processes require a significant investment if they were to be implemented in a smaller project, such as within agricultural datasets.

The datasets covered in Table 3.1 are largely created with annotation tools requiring manually drawing bounding-boxes or masks. There exists a number of works that aim to be an alternative or an improvement. Firstly, the process of drawing bounding-boxes or polygon masks can be cumbersome for annotators and a number of works have attempted to improve this process. For example, the bounding-box annotations from Open Images [33] is largely collected with extreme clicking [34], where the four most extreme points of an object are clicked from which a box can be determined. Alternatively, interactive annotation tools can propose annotations given a coarse input from a user. This is often done with an algorithm using scribbles or markings from a user highlighting foreground and background portions of the image [35–37]. Instead of altering the tool to create the required annotations, weak supervision aims to utilise weaker annotations to train a model for a more fine-grained task. Such an approach can be considerably cheaper as annotation is either faster or can be generated automatically. The training of the models can have algorithm improvements that are able to make assumptions about the location of a bounding-box or mask from annotations such as with points [38–40] or scribbles [38, 41]. Image level labels can also be combined with a strategy to select appropriate proposals [42, 43]. Recently

## Chapter 3. Understanding Neural Networks

**Table 3.1:** Overview of statistics and key processes for a number of object recognition benchmark datasets.

Dataset	Task	Object Classes	Annotated Images	Annotated Instances	Training	Roles	Stages	Gold Standard Sets	Annotator Location
ImageNet [13]	Bounding-boxes	200	516,480	534,309	Annotation Verification	Annotator Verifier	Category labelling Box annotation · Draw single box Box verification Coverage verification · pass: add image to dataset · fail: return image to box annotation	Verification training Box annotation	AMT
PASCAL VOC [30]	Bounding-boxes	20	22,531	27,450	Annotation	Annotator Observers Verifier	na	na	Researchers
COCO [5]	Instance masks	91	163,957	886,284	Annotation	Annotator Verifier	Category labelling Instance spotting · All instances from one class per image Instance segmentation · Single mask per iteration	na	AMT
LVIS [32]	Instance masks	1203	140,000	1,544,000	Annotation	Annotator Verifier	Category labelling · Max one object per image Instance spotting Instance segmentation · Single mask per iteration Segmentation verification Coverage verification	Instance spotting Instance segmentation Segmentation verification	Crowd-sourced
ADE20K [31]	Semantic masks	2693	22,210	434,826	na	na	na	na	Single annotator
Open Images [33]	Bounding-boxes Instance masks	600 350	1,910,098 997,947	15,851,536 2,785,498	Annotation	Annotator Verifier	Category labelling · Classifier model with human verification Box annotation · All instances from one class per image	Box annotation Instance segmentation	Internal (Google)
Objects365 [11]	Bounding-boxes	365	638,000	10,100,000	Annotation	Annotator Verifier	Category labelling Box annotation · All instances from one class per image	na	Crowd-sourced

self-supervision has become an increasingly popular approach to utilise unlabelled data. Self-supervision aims to learn representations that should transfer to another task by automatically generating labels [44]. A common approach in traditional supervised object detection is to finetune a model that is first pre-trained on a classification task, however the representations from this may not be well-suited to transfer to object detection. Self-supervised pre-training aims to determine more relevant representations for training a detector, by generating labels to train a model to predict the relative position between random crops [45], convert between grayscale and RGB in images [46], or predict the top proposals from a proposal algorithm [47]. Lastly, Semi-Supervised Learning (SSL) aims to learn a model from both a set of labelled and unlabelled data, where typically the labelled set is much smaller. This has mostly been popular in image classification compared to object detection. Primarily due to detection being a much more difficult task caused by class imbalance biases between foreground and background [48]. SSL approaches for object detection largely follow a teacher-student methodology of training one model using pseudo labels from another model which is trained on the labelled set. Pseudo labels can be prone to noise, making it difficult to train [48]. Recent approaches aim to address this using weighting techniques [48–53], heavy augmentations [48, 50–53], and consistency between multiple outputs [54].

It is clear that dataset creation is a critical task that has laid the basis for numerous advances in computer vision. Depending on the task at hand and the resources available, a practitioner must be aware of the requirements for manual annotation and consider adopting tools that can aid in the process.

### 2.3 Network Deployment

Earlier we covered a number of deep learning networks where the aim is to often to maximise performance on benchmark datasets. However, simply focusing on the optimal accuracy can come at the expense of increased hardware requirements to run the models. In an industrial context, there can be restrictions on the computing power available, especially in situations that require running the models at the camera. While in the field during harvesting, the roaming capabilities may not be available for cloud computing and an alternative is to be attached to the device, also known as at the edge. In recent years there has been focus for numerous companies on edge devices, such as NVIDIA Jetson [55] and Intel Neural Compute Stick (NCS) [56]. Edge devices typically have lower computational resources than workstations, therefore models are often optimised, such as with [57], but this can also potentially decrease the precision of the models. A number of works exist that aim to create lower complexity and high precision models that can be run on the edge [25, 58, 59]. Even with these networks care must to be

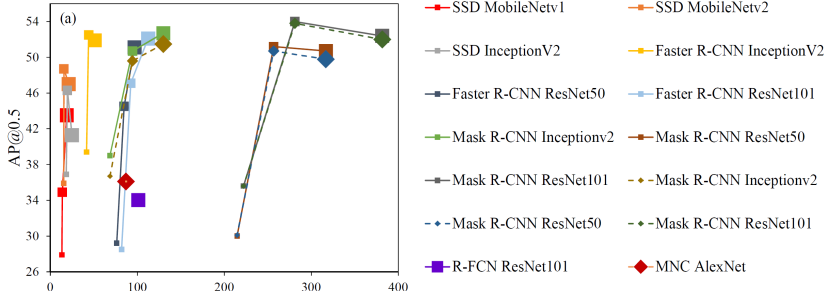
taken when developing models for deployment in the field and the choice of hardware cannot be neglected in favour of the optimal network architecture. The choice of hardware directly impacts the inference time, or in worse case, if the networks even can fit in memory.

### 3 Contributions

Understanding how neural networks work is a key point in deploying the systems into the real-world. Automating quality measurements, such as for WPCS, requires trust from farmers to allow this process to be completed by an AI-based model.

First, when developing computer vision systems it can be difficult to determine the model complexity required to capture the information needed in the model to solve the task. There are a large number of network architectures available in the computer vision community, each with their trade-offs in terms of accuracy, speed and model complexity. Therefore, we investigate this for our specific use-case of kernel fragment recognition using a large number of networks. We trained and evaluated three meta-architectures of different complexity in the form of the SSD [60], Faster R-CNN [9] and Mask R-CNN [61]. For each network we trained with feature extractors of increasing complexity, additionally, for each combination of meta-architecture and feature extractor, we trained models for three scales of image resolution. With a total of 28 models of varying complexity we were able to determine the trade-offs specifically for our task and propose an optimal architecture. In Figure 3.2 the trade-off is shown between inference time and AP at an Intersection-over-Union (IoU) threshold of 0.5 (AP@0.5). The AP@0.5 is evaluated against a hand-annotated test set of kernel fragments from three harvest seasons and the inference time is measured on an NVIDIA Titan XP. The meta-architectures and feature extractors can be seen in the legend and the image resolution is shown by different sized icons for each combination. Fast and well performing models are seen for a few SSD variants, however, an optimal trade-off is found for Faster R-CNN [9] with Inceptionv2 [62] at an image resolution of 400x730. This work is included in Paper D.

### 3. Contributions



**Fig. 3.2:** Trade-off in AP@0.5 and inference time for a number of meta-architectures and feature extractor for the task of kernel fragmentation. For each model three image resolutions are evaluated visualised by the decreasing size of the respective icon. Image adapted from [1].

To create a deep learning system for the task of WPCS quality an annotated dataset is required for training and testing of the networks. As covered earlier, high quality and consistent annotations is important and can be a key factor to the success of a project. Therefore, we have investigated the challenges of building datasets for deep learning in our context of WPCS. This was achieved by performing manual annotation with the aid of an annotation guideline as reference for annotators for both kernel and stover particles. The aim was to define rules such that all kernel fragments and stover overlengths were annotated in the image. In addition to guidelines, we have presented statistics and an evaluation of the quality of the dataset with respect to agreeance between annotators and expected instance sizes given machine settings. This evaluation found that despite best efforts, inconsistent annotations were provided between annotators and across harvest years. Therefore, an investigation into an alternative to pure manual annotation was conducted with SSL. It was found that competitive results in terms of AP and correlation analysis could be made when adding a large unannotated set of images. In Figure 3.3 example annotations are shown from the resulting dataset of polygons for kernel fragments (a) and stover overlengths (b). This work can be seen in more details in Paper E.



**Fig. 3.3:** Example annotations of kernel fragments (a) and stover overlengths (b). For kernels there is only a single class for the fragments, whereas for stover overlengths we define four classes for different parts of the plant.

We investigated the trade-off in model complexity for a specific task of kernel fragmentation, however, when deploying neural networks the final hardware has a significant role with respect to the speed the models can run. Furthermore, the price of the platform often has a role when deploying a system. Therefore, we investigated the trade-off when choosing an edge platform for a number of deep learning models for classification, object detection and semantic segmentation. For each task, we evaluated the speed from differing complexity and batch sizes for state-of-the-art networks. In addition to speed, the retail price of the platforms were considered in order to give an indication of best value when designing an edge-based computer vision system. In Figure 3.4 an example of evaluating the Frames per Second (FPS) and FPS cost for a lower complexity classification network is shown. Finally, we analysed the operations within the networks to identify which parts of the models could be optimised for further increasing speed. The work can be seen in more detail in Paper F.

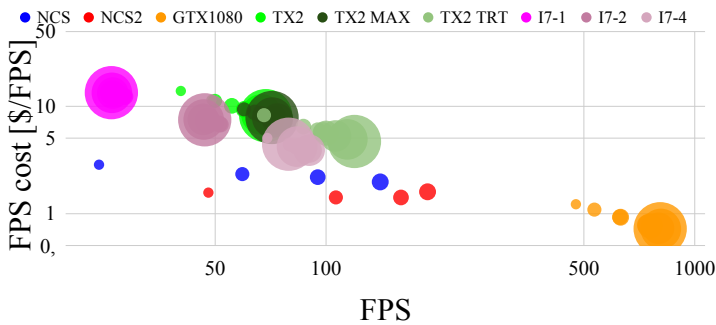


Fig. 3.4: The FPS cost for a low complexity deep learning classification network for a number of edge platforms. Image adapted from [3].

Our main scientific contributions across our papers within understanding neural networks can be summarised as:

- We document the trade-off in model complexity, speed and accuracy for the task of localising kernel fragments with a number of state-of-the-art object recognition networks.
  - Paper D: Evaluation of Model Selection for Kernel Fragment Recognition in Corn Silage.
- Present and discuss insights into the challenges of building datasets for deep learning in Whole Plant Corn Silage.
  - Paper E: The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage.

- Investigate the potential of adopting Semi-Supervised Learning for Whole Plant Corn Silage.
  - Paper E: The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage.
- We evaluate and document the trade-off in speed and price for three computer vision tasks on edge platforms and identified network operations for further optimisation aiding in edge-based system design.
  - Paper F: Evaluation of Edge Platforms for Deep Learning in Computer vision.

## References

- [1] C. B. Rasmussen and T. B. Moeslund, "Evaluation of model selection for kernel fragment recognition in corn silage," <https://arxiv.org/abs/2004.00292>, 2020.
- [2] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "The challenge of data annotation in deep learning – a case study on whole plant corn silage," *Under review at MDPI Sensors*, 2021.
- [3] C. B. Rasmussen, A. R. Lejbølle, K. Nasrollahi, and T. B. Moeslund, "Evaluation of edge platforms for deep learning in computer vision," in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds. Cham: Springer International Publishing, 2021, pp. 523–537.
- [4] 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.
- [5] 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*. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-10602-1\\_48](http://dx.doi.org/10.1007/978-3-319-10602-1_48)
- [6] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," 12 2016, pp. 379–387.
- [7] C. Rasmussen, "R-fcn object detection ensemble based on object resolution and image quality," 2017, unpublished Master Thesis, Aalborg University, Aalborg, Denmark.

## References

- [8] S. Bosse, D. Maniry, K. R. M. Iler, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *CoRR*, vol. abs/1612.01697, 2016. [Online]. Available: <http://arxiv.org/abs/1612.01697>
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [10] C. B. Rasmussen., K. Nasrollahi, and T. B. Moeslund., "R-fcn object detection ensemble based on object resolution and image quality," in *Proceedings of the 9th International Joint Conference on Computational Intelligence - Volume 1: IJCCI,, INSTICC*. SciTePress, 2017, pp. 110–120.
- [11] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8429–8438.
- [12] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Computers and Electronics in Agriculture*, vol. 178, p. 105760, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920312709>
- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "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.userix.org/system/files/conference/osdi16/osdi16-abadi.pdf>
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou,



## References

- and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [16] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [17] Z. Li, Y. Ma, Y. Chen, X. Zhang, and J. Sun, “Joint coco and mapillary workshop at iccv 2019: Coco instance segmentation challenge track,” 2020.
- [18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.
- [19] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, 2016.
- [20] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11263-013-0620-5>
- [21] L. Zitnick and P. Dollar, “Edge boxes: Locating object proposals from edges,” in *European Conference on Computer Vision*, September 2014.
- [22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3296–3297.
- [23] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” <https://github.com/facebookresearch/detectron2>, 2019.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, “SSD: single shot multibox detector,” in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905. Springer, 2016, pp. 21–37. [Online]. Available: [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)

## References

- [25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [26] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *CVPR*, 2019.
- [27] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.
- [28] D. Wang, M. Li, C. Gong, and V. Chandra, "Attentivenas: Improving neural architecture search via attentive sampling," *CoRR*, vol. abs/2011.09011, 2020. [Online]. Available: <https://arxiv.org/abs/2011.09011>
- [29] Z. Li, T. Xi, G. Zhang, J. Liu, and R. He, "Autodet: Pyramid network architecture search for object detection," *International Journal of Computer Vision*, 2021.
- [30] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," 2010.
- [31] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130.
- [32] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [33] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, pp. 1956–1981, 2020.
- [34] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4940–4949.
- [35] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, p. 309–314, aug 2004. [Online]. Available: <https://doi.org/10.1145/1015706.1015720>

## References

- [36] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [37] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [38] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 549–565.
- [39] L. Chen, T. Yang, X. Zhang, W. Zhang, and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8823–8832.
- [40] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Training object class detectors with click supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [41] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [42] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [43] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [44] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, "A survey of self-supervised and few-shot object detection," *arXiv preprint arXiv:2110.14711*, 2021.
- [45] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [46] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 649–666.

- [47] A. Bar, X. Wang, V. Kantorov, C. J. Reed, R. Herzig, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson, “Detreg: Unsupervised pre-training with region priors for object detection,” 2021.
- [48] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, “Unbiased teacher for semi-supervised object detection,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [49] Z. Wang, Y. Li, Y. Guo, and S. Wang, “Combating noise: Semi-supervised learning by region uncertainty quantification,” 2021.
- [50] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, “End-to-end semi-supervised object detection with soft teacher,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3060–3069.
- [51] Y. Tang, W. Chen, Y. Luo, and Y. Zhang, “Humble teachers teach better students for semi-supervised object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 3132–3141.
- [52] Q. Zhou, C. Yu, Z. Wang, Q. Qian, and H. Li, “Instant-teaching: An end-to-end semi-supervised object detection framework,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4081–4090.
- [53] H. Li, Z. Wu, A. Shrivastava, and L. S. Davis, “Rethinking pseudo labels for semi-supervised object detection,” 2021.
- [54] Q. Yang, X. Wei, B. Wang, X.-S. Hua, and L. Zhang, “Interactive self-training with mean teachers for semi-supervised object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 5941–5950.
- [55] NVIDIA, “Jetson nano developer kit,” <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>, 2021, accessed: 18 March 2021.
- [56] Intel, “Intel neural compute stick 2,” <https://software.intel.com/en-us/neural-compute-stick>, September 2020, accessed: 6 September 2020.
- [57] Intel, “Openvino toolkit,” [https://docs.openvinotoolkit.org/2018\\_R5/index.html](https://docs.openvinotoolkit.org/2018_R5/index.html), December 2020, accessed: 3 September 2020.
- [58] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *2018 IEEE Conference on Computer Vision and Pattern*

## References

- Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018.* IEEE Computer Society, 2018, pp. 4510–4520. [Online]. Available: [http://openaccess.thecvf.com/content\\_cvpr\\_2018/html/Sandler\\_MobileNetV2\\_Inverted\\_Residuals\\_CVPR\\_2018\\_paper.html](http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html)
- [59] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: <http://proceedings.mlr.press/v97/tan19a.html>
- [60] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Proceedings of the European conference on computer vision*, 2016, pp. 21–37.
- [61] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

## References

## Chapter 4

# Conclusion

This Industrial PhD has covered two topics, namely, monitoring corn silage quality and understanding neural networks. These were chosen in order to address the research question on whether the harvest quality monitoring could be automated with deep learning models on RGB images. Currently, for Whole Plant Corn Silage, there is no efficient and straightforward approach for determining the quality of the harvested silage. This is a crucial issue as harvesting with the correct machine settings is key to high quality fodder for dairy cows and ensuring high machine efficacy. We investigated measuring the kernel fragmentation and stover lengths in Whole Plant Corn Silage and found that current approaches require sieving and manual sample preparation. Therefore, we proposed to significantly speed up the measurement by using RGB images of non-separated Whole Plant Corn Silage. Through a number of scientific contributions we proposed different deep learning-based object recognition approaches for localising particles relevant to quality.

We investigated a number of two-stage recognition networks for the task of kernel fragmentation and stover chopping quality. The networks first find object proposals followed by box refinement and classification into either kernel fragments or four classes of stover overlengths. In these works we showed good performance in terms of object recognition metrics and correlation to sieved reference samples. For kernel fragments, a strong correlation for Corn Silage Processing Score was seen between physical measurements as well as scores estimated from annotations. Additionally, for stover overlengths, a strong correlation was found for two Theoretical Length of Cuts. The results from the models were found after researching strategies for data sampling, finetuning and anchor tuning in the Region Proposal Network, which significantly improved results compared to a naive training approach.

Further scientific contributions were made to estimating kernel fragmen-

tation quality with an efficient network for directly classifying relevant sieve sizes for particles. The classification step was achieved through algorithmic novelty with a sieve-based matching approach for object anchors. The novel network showed a considerable improvement on inference time compared to two-stage networks while having competitive correlation to Corn Silage Processing Score measurements.

As a focus in this PhD has been adopting deep learning models for automatic Whole Plant Corn Silage quality monitoring in the field it is important to understand how the neural networks perform. To this end, we have investigated the trade-offs in speed and accuracy in a number of state-of-the-art object recognition networks for kernel fragmentation. For three different meta-architectures we trained networks of varying complexity in feature extraction and differing image resolutions. For the models we could see a clear difference in inference speed and propose an optimal network architecture and image resolution for our task.

In order to train and test networks to localise non-separated kernel and stover particles we created a guideline for annotators on how to annotate. This was necessary in order to define the tasks, with the aim to annotate and thereby detect as many kernel fragments as possible to estimate the Corn Silage Processing Score. Whereas for stover, overlengths particles  $1.5\times$  greater than the Theoretical Length of Cut at the time of harvest should be annotated to estimate an Overlength Particle Score. Despite the guidelines we saw that annotators found challenges in the process as the agreement between them and sanity checks based on expected sizes varied. Regardless, we see promising results for the two tasks but recommend that care should be taken when evaluating models with metrics based on manual annotations. Furthermore, we investigated the potential of Semi-Supervised Learning as an alternative to training networks solely on manual annotations. We found promising results, indicating a potential usage of the method, especially when combined with a small number of annotated images from a single harvest.

For deployment of a deep learning-based system it is important to not only address model accuracy but also the platform it should run on. In situations where computations must be made on the device, such as in the field when harvesting, there can be computational restrictions due to power consumption and memory available. Therefore, for a number of edge platforms, we investigated the trade-off between speed, accuracy and price for three relevant computer vision tasks using deep learning models. For each model we evaluated the inference time in relation to the retail price which could help in determining the optimal platform based on a practitioners requirements on model complexity and budget. Additionally, we analysed the computation time for each of the models on the different edge platforms, giving indications on how to address further computational optimisation.



## Summary

The scientific contributions can be summarised to:

### Monitoring Corn Silage Quality

- The first works on estimating the quality of harvested corn silage in RGB images without the need for separating kernel and stover particles.
  - First presented in Paper A: Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Images.
  - Algorithm improvements in Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
  - Further improvements in Paper C: SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage.
- We propose robust two-stage networks for the task of localising kernel fragments and stover overlengths across harvest seasons and machine settings.
  - Kernel fragments evaluated in Paper A: Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Image.
  - Kernel fragments and stover overlengths localised in Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
- We show significant improvements to kernel and stover overlength quality monitoring in two-stage networks by investigating strategies for data separation and transfer learning, together with tuning parameters in the Region Proposal Network with respective shape and size characteristics for the two tasks.
  - Paper B: Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics.
- We present a novel sieve-based matching algorithm allowing us to train models for efficient estimation of kernel fragmentation quality.
  - Paper C: SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage.

### Understanding Neural Networks

- We document the trade-off in model complexity, speed and accuracy for the task of localising kernel fragments with a number of state-of-the-art object recognition networks.
  - Paper D: Evaluation of Model Selection for Kernel Fragment Recognition in Corn Silage.
- Present and discuss insights into the challenges of building datasets for deep learning in Whole Plant Corn Silage.
  - Paper E: The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage.
- Investigate the potential of adopting Semi-Supervised Learning for Whole Plant Corn Silage.
  - Paper E: The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage.
- We evaluate and document the trade-off in speed and price for three computer vision tasks on edge platforms and identified network operations for further optimisation aiding in edge-based system design.
  - Paper F: Evaluation of Edge Platforms for Deep Learning in Computer vision.

Through the work conducted in this PhD we investigated a number of deep learning networks for automatic quality monitoring of kernel fragmentation and stover overlengths in images of non-separated Whole Plant Corn Silage. In order to develop and deploy the neural network based systems we studied a number of areas in order to understand them. Based upon our contributions in these two themes, we believe that an efficient and straightforward Whole Plant Corn Silage automated monitoring system can be made using deep learning and object recognition. This is additionally highlighted by the fact that the company hosting the Industrial PhD has applied for two patents in Europe and the United States of America, indicating their belief in a future product.

Through our work and scientific contributions, the basis has been set for a system that can efficiently and robustly monitor a farmer's Whole Plant Corn Silage quality. This would result in higher earnings by increased yield, savings in regards to machine wear and lower fuel consumption. Furthermore, it places less requirements on the knowledge required by the forage harvester operator as specifics to quality are left to the algorithms.

## Future Work

This Industrial PhD has shown work on how to measure Whole Plant Corn Silage in RGB images of non-separated samples. However, before the models can be realised as a final commercial system there are a number of steps that should be addressed.

Firstly, previous measurement systems all rely on separating kernel and stover together with spreading out samples before an image is taken. This greatly decreases the difficulty and allows for the ability of determining precise contours for each particle. Such a precise measurement is naturally not possible in our images and we have not addressed any sources of sampling error that may be introduced by measuring Whole Plant Corn Silage directly. For example, particles occluding each other or particle orientation may result in underestimating the true size. While we evaluate against physical samples showing positive correlations, future steps can be taken to account for such errors, potentially improving the quality estimation. For example, a class could be added during annotation of a "covered particle" and models trained to detect these from which uncertainty can be addressed.

We spent extensive time on creating pixel-level annotations for our datasets. In addition, we attempted to include true population variances that a deployed system can be exposed to by using images from multiple harvests, different levels of crop maturity and harvester settings. The data used in this thesis was limited to Northern Europe and is smaller in comparison to considerably larger benchmark datasets often used in deep learning. Additionally, the physical sieving measurements are from two harvest weeks in a single season. Evaluating the system, either against annotated test sets or physically sieved samples, on more data with increased variance, can perhaps make the models more robust.

Our approaches for estimating kernel and stover quality vary. For kernels we aim to estimate the particle size distribution for all fragments, however only use overlenghts for estimating stover quality. Annotating all kernel fragments is an extensive task and places challenges on the networks in that smaller objects can be more difficult to detect. Therefore, using a similar approach to stover overlenghts by only localising kernel fragments which are too large could be of interest.

We investigated the trade-off for deep learning models on edge platforms in regards to speed and price. This is naturally also relevant when deploying the Whole Plant Corn Silage quality models. For the correlation analysis we estimated the quality on image sets consisting of a number of images, however, it is unknown what the minimum requirement on predictions and images are before a stable quality estimate can be made. Depending on this, together with a given edge platform, it can place a need for faster and less complex models than those evaluated in this thesis. If this is the case, the

trade-off in the quality correlation, complexity and speed should be investigated.

It would also be of interest to investigate if the results from our two-stage networks transfers to similar applications, both in agriculture and other industries. This could include other crops, for example, grains harvested from a combine harvester or before harvesting within localising relevant plants in crop rows. Additionally, the work presented on understanding neural networks for network architecture, dataset creation, and deployment can lead the basis for adopting our methods. Harnessing the power of deep learning by appropriate model development and understanding can hopefully digitise many areas of agriculture aiding farmers and continue to optimise food production.

## **Part II**

# **Monitoring Corn Silage Quality**



# Paper A

## Maize Silage Kernel Fragment Estimation Using Deep Learning-Based Object Recognition in Non-Separated Kernel/Stover RGB Images

Christoffer Bøgelund Rasmussen and Thomas B. Moeslund

The paper has been published in  
*MDPI Sensors* Vol. 19(16), pp. 3506, 2019.

© 2019 MDPI

*The layout has been revised.*



# Abstract

*Efficient and robust evaluation of kernel processing from corn silage is an important indicator to a farmer to determine the quality of their harvested crop. Current methods are cumbersome to conduct and take between hours to days. We present the adoption of two deep learning-based methods for kernel processing prediction without the cumbersome step of separating kernels and stover before capturing images. The methods show that kernels can be detected both with bounding boxes and at pixel-level instance segmentation. Networks were trained on up to 1393 images containing just over 6907 manually annotated kernel instances. Both methods showed promising results despite the challenging setting, with an average precision at an intersection-over-union of 0.5 of 34.0% and 36.1% on the test set consisting of images from three different harvest seasons for the bounding-box and instance segmentation networks respectively. Additionally, analysis of the correlation between the Kernel Processing Score (KPS) of annotations against the KPS of model predictions showed a strong correlation, with the best performing at  $r(15) = 0.88$ ,  $p = 0.00003$ . The adoption of deep learning-based object recognition approaches for kernel processing measurement has the potential to lower the quality assessment process to minutes, greatly aiding a farmer in the strenuous harvesting season.*

## 1 Introduction

Maize kernel processing evaluation is an important step in determining the quality of silage harvested from a forage harvester. Maize silage is used as fodder for cattle in dairy production and high quality silage though correct processing has an effect on milk yield [1] and suboptimal setting of the machinery can also lead to the quality being affected by up to 25% [2]. Kernels must be sufficiently cracked for efficient starch intake by lowering the requirement for chewing during eating and ruminating [3]. Kernels are processed by two mill rolls which compress and shear the plant. The gap known as the Processor Gap (PG) is often between 1 and 4 mm with 0.1 mm increments. This work focuses on the evaluation of kernel processing for silage quality efficiently through deep learning computer vision based methods via Convolutional Neural Networks (CNNs). Currently, the particle size distribution of kernel processing is evaluated through means which can be time consuming, cumbersome to conduct, and prone to error. An example of this is the Corn Silage Processing Score (CSPS) [3] and is one of the major standards in kernel processing evaluation. CSPS gives an analytical measurement of the kernel processing through laboratory equipment situated offsite typically returning a measurement after a number of days. In CSPS the user places a 160 g dried sample of harvested silage on a Ro-Tap sieving system which oscillates to allow processed kernels to pass through a number of differently sized sieve

screens. The materials that pass through a 4.75 mm sieve can be measured for starch content and the percentage of this that passes is the CSPS. Particles larger than this size may result in a slow starch digestion in cattle and increase chewing requirement. The CSPS can be interpreted according to [3] as greater than 70% is optimal processing, between 50% and 70% is adequate processing and less than 50% is considered inadequate processing. An additional finer sieve screen of 1.18 mm can be used to determine the number of over-processed kernels. The starch content in such fragments can simply pass through the cow's rumen, leading to wasted plant.

Another commonly used method for assessing kernel processing is the Penn State Particle Separator (PSPS) [4]. PSPS is similar to CSPS, however, does not require off-site laboratory equipment such as the Ro-Tap system or drying of the silage before starting the measurement process. Therefore, PSPS is able to give a farmer a much quicker indication of the kernel processing from the forage harvester. In PSPS three or four stacked trays with varying gaps are used to separate the kernel particles. The sample is placed in the top tray and the stack is shook a total of 40 times at a rate of one shake per second. After this, the weight of each tray is measured and is used to determine the distribution of kernel processing in the sample. Despite PSPS being more flexible than CSPS, the method is sensitive to the rate of shaking and moisture content, potentially giving a less accurate measurement.

The water separation method [5] can also be an effective method for a farmer to conduct a quick assessment of the kernel processing. Here, the total number of whole kernels in a 1-quart (946 ml) sample is evaluated. If more than one whole kernel per quart is found, the kernel processing is deemed not optimal. The method begins by placing the sample in a container filled with water. Then the sample is stirred gently until the stover, such as leaves and stalks, float and the kernels sink. Afterwards the stover and water is removed from which the number of whole kernels can be counted.

As mentioned, the aforementioned current kernel processing assessment methods are relatively time-consuming and can require potentially error-prone manual steps. There has been minimal work done in automating this process and to our knowledge only one such exists. In this work computer vision is used to calculate the kernel particle size distribution [6]. In the method, first kernels must be separated from the stover using a method such as water separation. After this, the kernels are placed without touching any other samples on a dark background together with a common coin whose size is known, such as a penny. The coin can then be used as a reference later on in the system to calculate the kernel sizes. An image is captured and the contours of the kernel particles are found via image processing. Then the maximum inscribed circle is found for each particle in pixels which is converted to a kernel particle size distribution in millimetres. Metrics such as the percentage of particles smaller than 4.75 mm or average area give an

indication to the user of kernel processing quality.

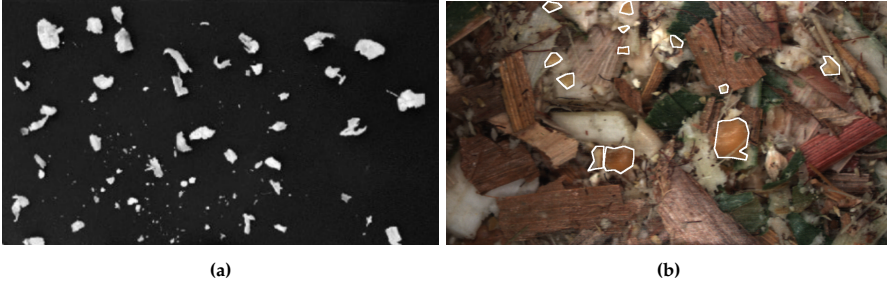
Looking into the broader domain, there is a large amount of research into measuring the quality of other crops. Firstly, the grades of product are determined by calculating rice kernel shape and size features and training a support vector machine [7, 8]. Additionally, in [9] rice colour features and Fourier descriptors for shape and size are extracted from which the quality grade is determined through multivariate statistical analysis. A number of methods identify whole or broken fragments in grains. In [10], the size, color and brightness values are used in combination with a flatbed scanning device. In [11], rice is segmented based on color, and shape features indicate the grade of the crop. Classification of the grains in the image can be necessary when different grain types are mixed. Artificial neural networks have been used to classify types based upon extracted handcrafted features. In [12] color and texture features, in [13] size, color, and shape features, and in [14] color and morphological features were used to train networks respectively. K-Nearest Neighbor classifiers were trained on size and texture features in [15, 16], with a number of color models being used in the latter. The quality of maize seeds was evaluated in [17] using hyperspectral imaging where data was reduced through t-distributed stochastic neighbourhood embedding and Fischer's discriminant analysis for quality classification.

The works mentioned so far all follow that traditional computer vision approach of extracting hand-crafted features followed by using a classifier to make a decision on the task at hand. However, since 2012 when AlexNet [18] won the ImageNet classification challenge by a significant margin, deep learning with CNNs has dominated the field. Object recognition in images is a challenging task due to potential variations in objects, such as the colour, texture, shape, and size, and variations in images, such as the lighting, viewpoint, and occlusion. CNNs have been shown to learn complex patterns in data through a hierarchy of layers. Typically earlier CNN layers capture simple patterns such as the edges, while later layers learn more complex representations such as the shape of specific objects. This hierarchy has the potential to learn a powerful model given high quality data. There are numerous examples of machine vision with deep learning in agriculture that show good results and in many cases a significant improvement over using hand-crafted features. Examples include [19], where fully convolutional neural networks were trained to predict a semantic segmentation map of clover, grass, and weeds in RGB images containing clover-grass mixtures to estimate the distribution of the classes in the field. Here, they account for the potentially large amount of training data required for CNNs, as it was observed the annotation could take up to 3.5 h for 10 images. New images were simulated by combining augmented objects from those already annotated on top of captured background images. A deep learning approach to detect tomato plant diseases and pests was done in [20], where a number of popular models

was evaluated for the task. In [21] a CNN and random forest classifier was trained to classify 32 different species of leaves. Plant disease detection of 14 different crop species including 26 diseases was done in [22] using CNNs and a number of different feature extractors such as AlexNet [18]. Crop and weed detection using CNNs was done in [23] on a combination of RGB and near-infrared data.

The aim of this work is to create a system to localise kernels fragments in RGB images for kernel processing assessment without the requirement separation of stover and kernels such as in [3, 4, 6]. Such a system will allow the farmer to gain an insight into the quality of the kernel processing without the need to perform a time-consuming and cumbersome process. We propose to train CNNs in both a bounding-box detector and instance segmentation form to automatically detect and localise kernel fragments in the challenging images. Examples of the images used in this work are shown in the following section in Figure A.3. The methodology in training the aforementioned networks will be covered in Section A.2 and the achieved results in Section A.3.

An example of the difference between separated kernel/stover images such as those typically used in [6] and non-separated used in this work can be seen in Figure A.1. Additional white outlines in Figure A.1b represent the outline of kernel fragments.



**Fig. A.1:** Example of the difference in images between separated and non-separated corn silage. (a) Reprinted from [6], with permission from Elsevier; (b) Example image from this work.

## 2 Materials and Methods

This section details the materials and methods used in the work. This includes images, subsequent kernel annotation and overview of the CNN models and training parameters. In order to train the respective recognition algorithms, a dataset of harvested silage is required. Both the basis for the images of the silage and annotation with resulting datasets is covered.

## 2. Materials and Methods

### 2.1 Images

RGB colour images were taken of harvested silage over three years. The silage was produced from a variety of fields and crop conditions, and harvested with different machine settings. For example, the PG primarily accounts for the differences in the level of kernel fragmentation by altering the distance between two rollers mills in which the corn plant passes through. Secondly, the cutting length (CL) affects how fine the corn plant is chopped before passing through the rollers. Figure A.2 shows an example of harvested silage, while the two images in Figure A.3 show the differences in the harvested silage and a small PG (a) and a larger PG (b), resulting in a higher proportion in smaller and larger kernel fragments respectively. The silage in both (a) and (b) were harvested with same CL. Additionally, in the images a scale is shown in the bottom right indicating 1 cm, which equates to a resolution of 0.05 mm per pixel.



Fig. A.2: Example of harvested silage.



Fig. A.3: Example images of the differences in silage harvested with varying fragmentation. The white outline shows kernel fragment annotation outlines. (a) Smaller Processor Gap (PG) resulting in smaller kernel fragments; (b) Larger PG resulting in larger kernel fragments. A scale in the bottom right of the images shows the size of the images where 200 pixels is equal to 1 cm.

## 2.2 Datasets

The images were annotated using a tool with user defining vertices outlining the kernel fragments creating a polygon for each instance in a given image. These vertex-based annotations can be used to train both the instance segmentation models or they can be converted to bounding-boxes by taking the outer extremas of the annotated vertices for detection models. Just under 2500 images were annotated across the data collected from three years, with the largest number of annotations being done on the images collected in 2017 as seen in Table A.1. It is also shown in the table that a total of four datasets were created, one for each of the harvest years (2015, 2016, & 2017) and a final set that contains all of the data from the three years combined (151617). For each of the datasets, train and test is split randomly at roughly 60% and 40% respectively. The division of years was done to evaluate how a CNN model would react to being trained on images from one harvest with its given conditions and how the resulting model would perform on images from another harvest year. The visual appearance of the crop can change due to the variations in farming such as geographical location, weather conditions, or plant maturity. The combination of data in 151617 is to evaluate the large data requirement of deep learning models and to see if models tuned to specific conditions or a model with larger variation is preferable.

**Table A.1:** Overview of datasets created based on the year in which the images were captured. The total number of images and kernel instances per dataset is shown.

	2015	2016	2017	151617
Train Images	111	115	1167	1393
Train Kernel Instances	1388	675	4844	6907
Test Images	76	85	884	1045
Test Kernel Instances	836	433	3425	4694

## 2.3 Deep Learning Models

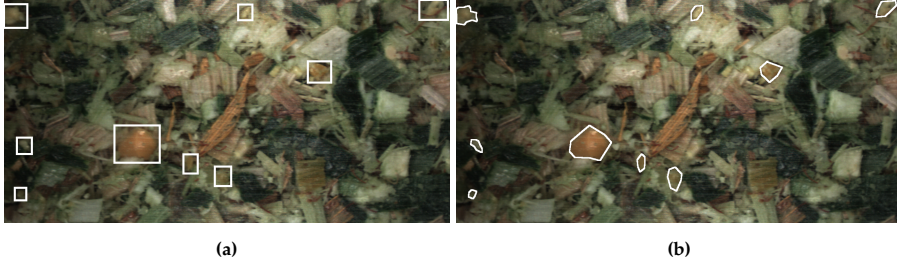
This section covers the two deep learning approaches used for kernel fragment recognition in both object detection and instance segmentation form. First, we will give a short overview of deep learning and CNNs with respect to the core concepts. Deep learning is a form of machine learning that aims to solve a task using a “deep” model through the transformation of data using various functions that can represent the data in a hierachical manner [24]. Deep learning can be especially successful as it allows for automatic feature extraction, rather than an engineer designing hand-crafted features. If the dataset is representative of the deployment scenario it can allow the model to learn a strong set of functions that can be difficult for an engineer

to find. However, due to this deep hierarchical manner, the features determined by the model can be difficult to debug and are often treated as a black box. In deep learning the aim is to have model learn a feedforward mapping between input and output, for example, given an input of an RGB image of maize silage output, the x-y coordinates of kernels together with a confidence score of the prediction. In order to learn this mapping the aim is to update the parameters of the model through training to give the desired output. The model is trained over a number of iterations where given the model and its current set of parameters, it perform the feedforward mapping for an image and measures the error of the model in comparison to the correct answer defined in the annotation. This error can then be used to push the model parameters in the correct direction by updating them through the method of backpropagation. Here, the error traverses back through the network and computes the gradient for each function's parameters that should decrease the error. Using this gradient, an optimisation algorithm, such as Stochastic Gradient Descent (SGD), updates the parameters of the function. This process is continuously performed until the model has updated the parameters in such a way to best perform the mapping of input and output with the lowest possible error in the training set whilst still performing well on a validation set. Depending on the task there are a number of different types of architectures within deep learning: this includes recurrent neural networks often used for natural language processing, reinforcement learning used in robotics, and CNNs used in this case for RGB images. With CNNs the deep hierarchy of functions mainly revolve around the convolution mathematical operation which is well suited for the grid-like topology of images. The convolution operation is relatively simple and has been used in hand-crafted feature engineering such as edge detection or image blurring. Convolution is computed by a filter of a given size (i.e.,  $3 \times 3$  or  $5 \times 5$ ) sliding over the image data and computing an elementwise multiplication and producing a single output value in a feature map. Convolution over the entire image produces a fully realised feature map. The deep aspect of CNNs is therefore a large number of convolution layers computing feature maps upon previously computed maps in succession. The learning process described earlier for CNNs aims to learn the weights in the hierarchy of convolution filters that give the optimal mapping between input and output.

The methods chosen in this work are the Region-based Fully Convolutional Network (R-FCN) [25] for bounding-box detection and the Multi-task Network Cascade (MNC) [26] for instance segmentation. These were chosen due to their state-of-the-art nature at the time of conducting this work, where both performed well on a number of object recognition benchmarks including PASCAL VOC [27] and MS COCO [28].

The CNN approaches solve the task of object recognition but at different degrees of localisation granularity. Bounding-box detectors place an axis-

aligned bounding-box around the detected object whereas segmentation indicates the object at a pixel level. Due to the lower localisation granularity of bounding-box detectors, they may over-sample the object and give a larger indication of size than is actually true. This difference on an image from this work can be seen in Figure A.4.



**Fig. A.4:** Examples of the difference in localisation granularity between bounding-boxes and segmentation. (a) Bounding-box localisation, (b) Segmentation localisation. The segmentation localisation fits much closer to the kernel instances and thereby can give a more precise measurement on kernel size.

This remainder of this section includes an overview of how the methods perform their respective forms of object recognition by covering the model architecture and defining the model and learning parameters used in this work.

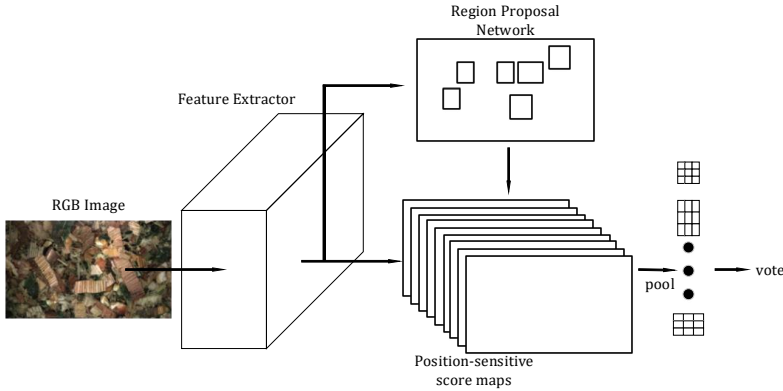
### Region-based Fully Convolutional Networks (R-FCN)

R-FCN is a bounding-box CNN-based object detection method and is based on the popular two-stage detection strategy of object proposals followed by classification of found proposals. Additionally, the authors were one of the first to adapt Fully Convolutional Networks (FCNs) into the two-stage pipeline, rather than using feature pooling layers, such as Region of Interest (RoI) pooling as in the Faster R-CNN detector [29]. Thus, potentially important spatial information is not discarded as can be the case when pooling features. The R-FCN architecture can be seen in Figure A.5. In the first stage, an input RGB image is passed through a number of convolutional layers to create a deep representation through a number of feature maps. As is common practice in object recognition through CNNs, the convolutional layers can take many forms that can vary in complexity. Popular choices for the layers include AlexNet [18], VGG [30] and ResNets [31], where in the original R-FCN work the ResNet-101 network was primarily explored. Class-agnostic RoI object locations are found by a Region Proposal Network (RPN) [29]. The RPN finds RoI proposals by sliding a small network over the last feature map computed by the previous convolutional layers. At each sliding win-



## 2. Materials and Methods

down location a number of anchor boxes with varying scales and aspect ratios predict the confidence of a location containing an object. In the second stage, candidate RoI proposal features via an FCN for classification are extracted from a number of position-sensitive score maps. A total of  $k^2(C + 1)$  maps are computed where  $C$  is the number of object classes and  $k^2$  is the spatial grid representing relative positions. In the case shown in Figure A.5,  $k = 3$ , therefore, nine score maps are computed for each object class.

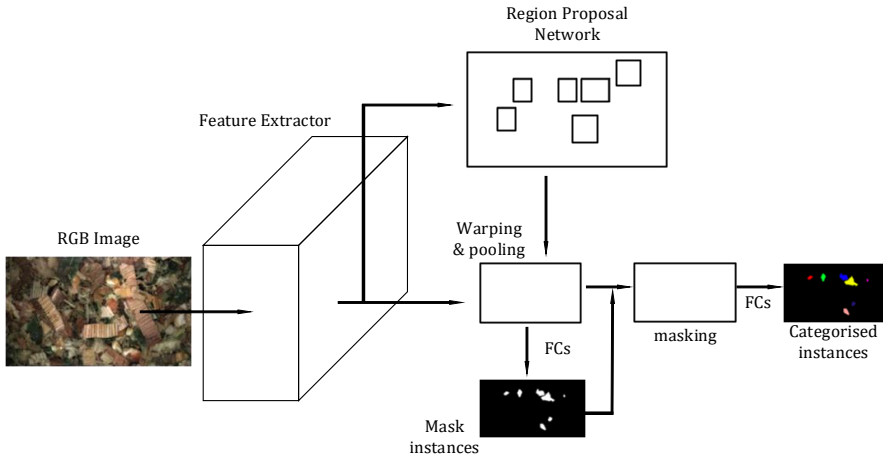


**Fig. A.5:** The R-FCN architecture illustrating an image being passed through a number of convolutional layers. RoIs are computed from a RPN on the final convolutional layer, these RoIs are classified through the coloured position-sensitive score maps.

The R-FCNs trained for kernel detection in this work largely follow the same procedure as that conducted in the original work. The network weights were initialised from a pretrained ResNet-101 for ImageNet [32] classification supplied by the authors. The networks were trained for a total of 110,000 iterations using SGD with an initial learning rate of 0.001 and after 80,000 iterations the learning rate was dropped by 0.1. Additionally, momentum of 0.9 and weight decay of 0.0005 was used during optimisation. With respect to the position-sensitive score maps  $k = 3$ . For each image the mean RGB ImageNet values are subtracted to normalise the training set which aids in the learning process. Subtracting the mean RGB from our training datasets was also evaluated during early development, however, it showed that results were better when using the ImageNet means. Horizontal flipping was the only data augmentation strategy used during training and images were scaled such that the height was 600 pixels and the width was then scaled accordingly to keep the original aspect ratio.

### Instance-Aware Semantic Segmentation via Multi-Task Network Cascades (MNC)

MNC also follows the mantra of multi-stage object recognition. The task in MNC is instance segmentation where the key difference between R-FCN is a module for determining mask instances, in addition to the region proposals and classification modules. The general architecture of MNC can be seen in Figure A.6. As in R-FCN, a feature map is extracted from the last of a number of convolutional layers computed based on an input RGB image. The authors performed their primary experiments using the VGG-16 networks, however, as in R-FCN any popular or user-designed CNN architecture can be used for feature extraction. An RPN determines class-agnostic region proposals followed by RoI warping and pooling. These are used as input to the mask generation modules in combination with learnt fully-connected (FC) layers. Finally, the masks in combination with another set of FC layers perform classification of the mask instances.



**Fig. A.6:** The Multi-task Network Cascade (MNC) architecture, as in Region-based Fully Convolutional Network (R-FCN), an image is passed through a number of convolutional layers and RoIs are found with an RPN. Features are extracted from the RoIs via RoI warping and pooling. Class agnostic masks are found from the features that are being passed through FC layers. The masks are classified from the RoI features through another set of FC layers.

As the name implies and as shown in Figure A.6, MNC is a cascaded approach for instance segmentation of first determining box instances then mask instances and lastly categorising the instances. However, it is common practice to refine the predictions by extending the cascade to five stages by repeating both the mask generation and classification module. This approach was adapted in this work from the open source code provided by

the authors. The work included a pre-trained VGG-16 network trained on ImageNet which was used for transfer learning. However, due to the large complexity of using VGG-16 as a feature extractor, an ImageNet pre-trained AlexNet feature extractor was adapted instead. Following the author’s procedures, MNC models were trained for a total of 25,000 iterations using SGD with an initial learning rate of 0.001. After 20,000 iterations, the learning rate was decreased by 0.1. Additionally, momentum of 0.9 and a weight decay of 0.0005 was used. As in the R-FCN models, the ImageNet RGB mean values were subtracted from the images. Again, horizontal flipping was the only data augmentation implemented and images were scaled such that the height was 600 pixels with width scaled accordingly.

## 2.4 Hardware

Models were trained on an Ubuntu 16.04 machine with an NVIDIA Titan XP Graphics Processing Unit (GPU) using the Caffe framework [33]. Caffe is a deep learning framework developed by Berkely AI Research that allows for fast training of testing of multiple types of models including CNNs and recurrent neural networks. An overview of the memory requirements for training the R-FCN and MNC models and inference speed can be seen in Table A.2. While the two models have a relatively low requirement on GPU memory, the difference in the feature extractor can be seen for both train and test memory. The considerably larger and more complex ResNet-101 model present in R-FCN increases the memory usage and adds to the inference timings in comparison to MNC with the AlexNet backbone.

**Table A.2:** Overview of hardware statistics for both methods. Timings were done on images of size  $600 \times 1000$  pixels on an Ubuntu 16.04 machine with an NVIDIA Titan XP GPU.

	Train Memory (MB)	Test Memory (MB)	Inference Time per Image (s)
R-FCN (ResNet-101)	6877	3251	0.101
MNC (AlexNet)	3439	2369	0.087

## 2.5 Computer Vision Metrics

Both of the algorithms can be evaluated on an object-level. These metrics do not directly measure how well a prediction intersects with the ground truth instance, rather, it is a measurement of whether or not an instance is correctly classified given a minimum Intersection-over-Union (IoU) threshold between the two. If a prediction overlaps by more than the IoU threshold it can be determined as a true positive detection, otherwise, it is a false positive. In this work an IoU of 0.5 is used when presenting results for the object-level metrics. It should also be noted that only a single prediction can be

considered as a true positive with a given ground truth—typically this is the prediction with the highest IoU. If multiple predictions overlap above the threshold, the remaining are considered as false positives.

Firstly, the precision on a dataset can be calculated as:

$$Precision = \frac{TP_{objects}}{TP_{objects} + FP_{objects}}, \quad (A.1)$$

where  $TP_{objects}$  and  $FP_{objects}$  are the total number of true positives and false positives object instances.

The recall of a dataset is calculated by:

$$Recall = \frac{TP_{objects}}{P_{objects}}, \quad (A.2)$$

where  $P_{objects}$  is the total number of positive ground truth examples.

Additionally, Average Precision (AP) is calculated as the mean precision of a dataset and is calculated across 11 equally spaced levels of recall [0, 0.1, ..., 1]. AP is determined by:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} \rho_{interp}(r), \quad (A.3)$$

where the precision at each level of recall  $r$  is interpolated by the maximum precision measured for which the corresponding recall exceeds  $r$ :

$$\rho_{interp}(r) = \max_{\tilde{r}: \tilde{r} \geq r} \rho(\tilde{r}), \quad (A.4)$$

where  $\rho(\tilde{r})$  is the measure precision at recall  $\tilde{r}$ .

The F1-score is calculated by:

$$F1-Score = \frac{2TP_{objects}}{2TP_{objects} + FP_{objects} + FN_{objects}}, \quad (A.5)$$

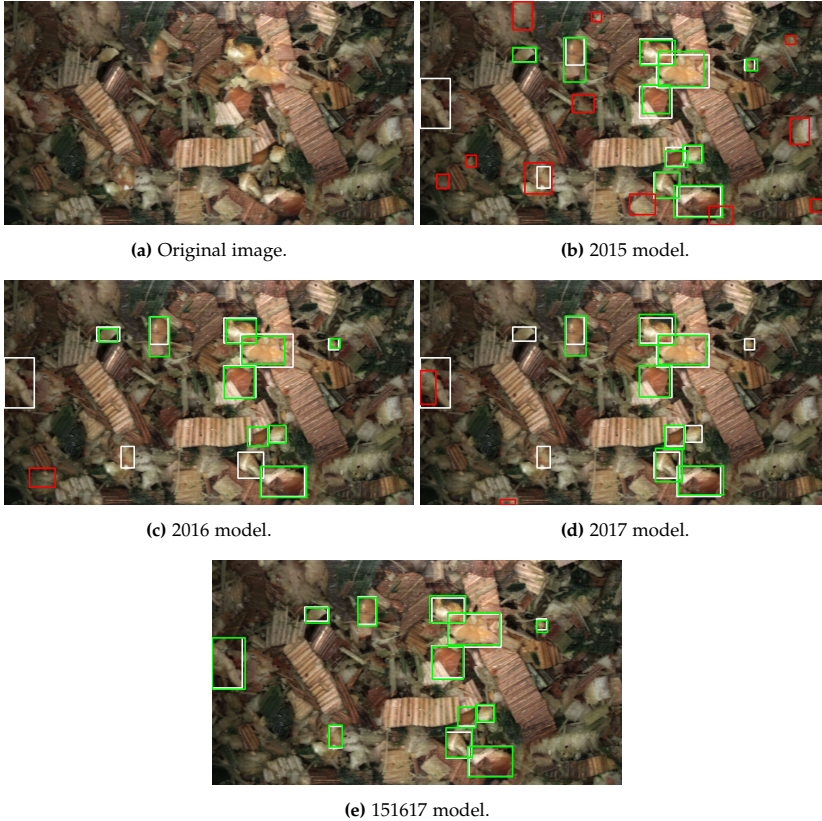
where  $FN_{objects}$  are the total number of non-identified ground truth instances.

### 3 Results

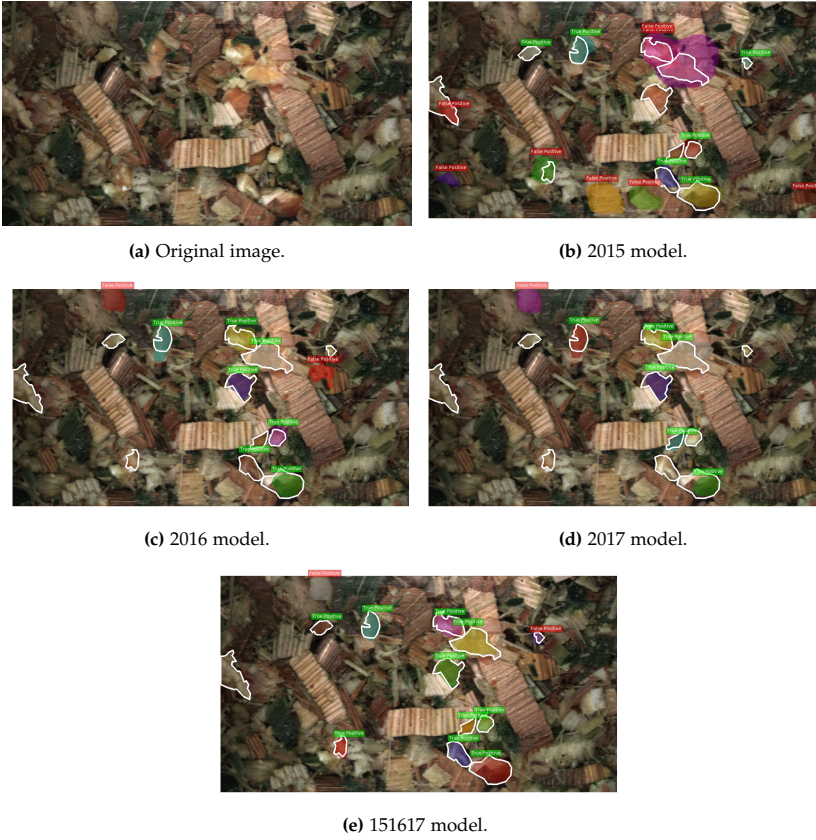
The results for the various trained models according to the metrics defined in Section 2.5 will be covered. Finally, an analysis of Kernel Processing Score (KPS) will be conducted to address the potential of using the system for silage quality evaluation in an industry setting.

### 3.1 Computer Vision Results

Firstly, detections from the four trained R-FCN models can be seen on an example image from the 2016 test set in Figure A.7 and the corresponding four MNC models in Figure A.8. In Figures A.7b–e the ground truth bounding-box annotations are shown in white around the kernel fragments, true positive detections are shown in green, and false positives are shown as red. Whereas in Figures A.8b–e the annotations are shown as a white outline around the kernel fragment, individual kernel fragment instance predictions are shown with different colours, while the determination of true positive or false positive is indicated by the green or red text above the prediction. In both figures, detections were considered as either a true positive or false positive at an IoU threshold of 0.5. The original image can be seen in Figure A.7a and Figure A.8a.



**Fig. A.7:** Model predictions on a test image from 2016. Bounding-boxes colours indicate ground truth (**white**), true positive (**green**) and false positive (**red**). True positives and false positives evaluated at an IoU threshold of 0.5.



**Fig. A.8:** Model predictions on a test image from 2016. Ground truth annotations are shown as a white outline around the kernel fragment. The colour in the text box indicate true positive (**green**) and false positive (**red**). True positives and false positives evaluated at an IoU threshold of 0.5. The individual colour for each prediction indicate separate instances of predictions.

An overview of the metrics covered in the previous section are shown below for the models trained and tested on the respective datasets defined in Table A.1. As stated in Sections 2.3 and 2.3, the four respective R-FCN and MNC models were trained using a consistent architecture and learning parameters. The only difference is the training dataset itself, where the content aimed to give an insight into the varying field conditions in agriculture from harvesting season to season. Additionally, there is a considerable difference in the amount of data annotated in the sets, where the 2017 sets have around  $10\times$  more images in both training and testing. Of course, this also has the effect of images captured in 2017 being the significant majority in the combined 151617 dataset. An overview of the results for the computer vision metrics can be seen in Table A.3. For each test set the best performing model for a given metric is shown in bold. The general trend seen in the table is that

### 3. Results

the 151617 model is the most robust across the four test sets, in many cases being the best performing for a metric or the second best. The differences between then R-FCN and MNC models are slight with only a few percentage points difference for all test sets apart from the 2015 test set. For this test set the model trained on the larger 151617 dataset performs considerably better than the other models across all metrics, including the 2015 model which is trained on only images from the same year as the test set. The R-FCN 151617 model achieves a 65.9% AP, 31.9% points higher than that of the 2015 counterpart. Whereas the AP for the MNC model is significantly lower at 40.4%, a significant increase is still present compared to the 2015 MNC model. Additionally, for the 151617 R-FCN model precision and recall scores at 70.0% and 76.0%, roughly 20.0% points higher than the 2015 model in both regards. The considerable improvement of the 151617 model in comparison to 2015 is present despite images from 2015 only making up around 10% of the training material in 151617. However, this 10% in addition to the roughly 10% from 2016 seems to have a significant impact as the model trained on data only from 2017 performs worse than both 2015 and 151617 models at 28.5% AP.

**Table A.3:** Computer vision metric results for both the R-FCN and MNC models across the four test sets.

Train Dataset	R-FCN				MNC			
	AP	Prec	Recall	F1-Score	AP	Prec	Recall	F1-Score
<b>2015 Test</b>								
2015	34.0	55.5	53.0	54.2	27.7	44.5	31.8	37.1
2016	19.0	<b>80.0</b>	21.0	33.3	16.8	60.5	16.5	25.9
2017	28.5	51.1	40.2	45.0	27.7	50.0	32.1	39.1
151617	<b>65.9</b>	70.0	<b>76.0</b>	<b>73.9</b>	40.4	50.3	46.3	48.2
<b>2016 Test</b>								
2015	25.3	23.3	87.1	36.8	40.7	30.1	61.0	40.3
2016	41.8	52.1	73.2	60.9	52.1	54.4	62.8	58.3
2017	34.2	41.7	63.1	50.2	53.0	45.7	67.9	54.6
151617	66.9	<b>56.9</b>	<b>90.8</b>	<b>70.0</b>	<b>71.8</b>	47.6	80.8	59.9
<b>2017 Test</b>								
2015	15.3	19.0	<b>70.5</b>	29.9	18.6	20.2	36.4	25.8
2016	19.2	<b>43.4</b>	44.1	43.7	24.3	39.8	32.8	36.0
2017	31.0	36.4	66.9	47.2	<b>36.3</b>	32.9	53.3	40.7
151617	33.4	37.6	<b>67.2</b>	<b>48.2</b>	35.9	31.9	53.7	40.0
<b>151617 Test</b>								
2015	19.6	23.4	<b>73.6</b>	35.6	26.1	26.2	42.9	32.5
2016	22.3	<b>50.1</b>	44.7	47.2	28.4	46.7	34.2	39.5
2017	30.2	39.2	62.5	48.2	35.8	36.0	51.0	42.2
151617	34.0	40.7	66.0	<b>50.4</b>	<b>36.1</b>	34.2	52.2	41.4

As mentioned, the difference in results between R-FCN and MNC models are not as significant for the remaining test sets, however, the trend of the combined 151617 training dataset giving robust results continue. The 151617 models is the best performing for both models by considerable margins. AP for the 151617 model scores at 66.9% and 71.8% for R-FCN and MNC respectively, 25.1% and 19.7% points higher than the 2016 models. Similar increases in the remaining metrics exist as of that for the 2015 test set. Once again images similar to the test set is in the minority in the 151617 training set with around 10% being harvested in 2016. As in the results for the 2015 test set this 10% addition has a considerable effect as the relatively large 2017 model is the third best performing model on most metrics.

The 2017 and 151617 results do not show an as significant difference in the results as for 2015 and 2016. The best performing model varies across the numerous metrics, however, the 2017 and 151617 models measure consistently well in comparison to the other two who lack in some regards. For example, the 2015 R-FCN model has a relatively high recall of 70.5% but poorer precision of 19.0%. Whereas the 2016 R-FCN model has the highest precision on both 2017 and 151617 test sets at 43.4% and 50.1%, however, the AP is considerably lower at around 10% points. The results are similarly not as varying for the MNC models, with the 2017 and 151617 models in general performing strongest. In general, there is negligible difference between the 2017 and 151617 models for both R-FCN and MNC on the corresponding two test sets. This is likely because the training set between the two models has much more overlap than the earlier results.

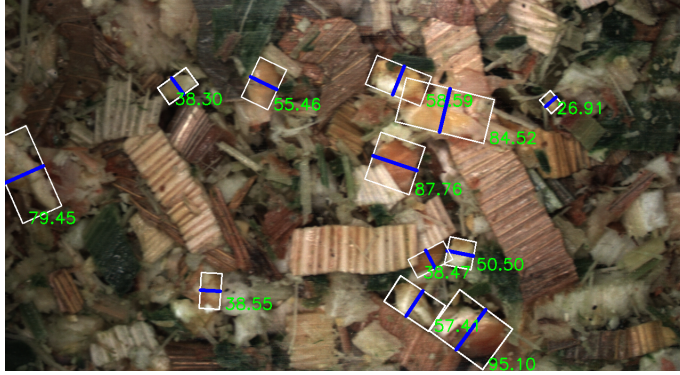
### 3.2 Kernel Processing

To evaluate the viability of the two CNN methods for kernel fragment recognition, we adopt the commonly used KPS score from the CSPS [3]. For each detected instance for either method the length of smallest axis from a rotated fitted bounding-box is found. This length gives an indication of the detected kernel instance that would pass through the 4.75 mm sieve screen used in CSPS. The smallest axis length is used as a quality indicator due to the three-dimensional shaking present in the Ro-Tap separators used in CSPS, therefore, particles are separated based upon the shortest diameter. The KPS was also used to evaluate the image processing algorithm developed in [6], however, the diameter of the largest inscribed circle was used in this case. Additionally, in [6] the actual KPS was calculated by performing the Ro-Tap laboratory separation, unfortunately, this was not done while harvesting in this work. Instead we calculated the KPS for a sequence of images from a given PG by determining the shortest axis length from the ground truth annotations. As the images of the silage were taken with known distance to the camera, the pixel resolution in mm can be converted as 1 mm to 20 pixels,



### 3. Results

meaning that kernels of lengths below 95 pixels (4.75 mm) are considered to be optimally processed. Figure A.9 shows the calculation of the minor axis from a rotated bounding-box for an annotated image. In this example a single kernel is above the 4.75 mm threshold and deemed not optimally processed with a minor axis of 95.10 pixels.



**Fig. A.9:** Visualisation of determining kernel processing based on the shortest axis length of a rotated bounding-box for a number of annotated kernel fragments. The shortest axis is shown via a blue line with the length in pixels for each shown next to the fragment.

Due to the large number of annotations present for images from 2017, a number of different sequences were created with different conditions. This is shown in the left-most two columns in Table A.4, with 17 sequences with varying PGs. The table also shows the KPS calculated as the percentage of kernel fragment detections with a shorter axis below 4.75 mm for the eight respective models trained on different subsets of data. It should be noted that because of the nature of the predictions between the R-FCN and MNC models, it was only possible to determine a rotated bounding box for the MNC predictions due to the higher localisation granularity of pixel-level segmentation. Instead, for the R-FCN detections, the shortest distance of the axis-aligned bounding box was taken. Finally, the KPS ground truth from the annotations is shown in the right-most column. The average absolute error summarises the accuracy of each model of all PGs in the final row. While there are individual differences in the KPS calculation in comparison to the annotations from different sequences, in general the average absolute error is lowest for the 151617 R-FCN model—4.5% points less than the MNC counterpart despite having the disadvantage of axis-aligned bounding-boxes.

**Table A.4:** Kernel Processing Score (KPS) results across sequences of varying PGs for R-FCN and MNC models. The final row shows the average absolute error for each model over all sequences from the 2017 test set.

% (<4.75 mm)	2015		2016		2017		151617		
PG	R-FCN	MNC	R-FCN	MNC	R-FCN	MNC	R-FCN	MNC	Annotation
1	96.2	97.7	91.8	94.7	93.8	95.5	92.2	97.3	93.5
1	95.4	95.4	95.2	96.1	95.8	98.8	95.1	97.7	98.7
1	88.0	76.4	85.7	86.5	81.3	87.2	80.7	88.9	79.9
1	93.7	94.8	93.0	91.1	92.5	95.7	92.1	96.1	94.3
2	93.9	94.8	78.8	75.2	89.2	95.8	87.8	97.3	79.1
2	92.8	97.7	89.9	92.6	86.3	95.7	90.8	95.7	93.8
2	84.8	71.5	84.2	100.0	82.5	85.8	82.7	87.7	88.8
2	88.0	86.1	86.4	85.6	82.2	90.6	76.0	92.6	79.1
3	89.6	80.7	85.1	83.5	82.4	89.3	81.8	90.4	82.3
3	94.6	95.2	91.2	95.7	89.9	94.1	86.1	93.8	85.7
3	90.4	85.9	83.2	83.1	77.9	90.3	80.5	90.0	79.3
3	89.1	86.3	83.6	84.5	88.5	93.0	89.8	91.8	94.5
3.5	90.2	80.8	83.5	88.0	81.4	89.5	81.2	91.2	83.1
3.5	88.6	75.5	84.0	81.6	79.7	89.0	80.3	90.2	76.6
3.5	91.2	93.0	89.5	92.6	91.7	93.2	92.9	94.6	92.4
3.5	85.6	75.9	75.4	72.5	79.5	84.7	78.4	86.1	73.7
3.5	91.5	89.8	86.8	91.3	86.6	91.5	86.9	92.9	86.4
Avg. abs. error	6.7	5.3	3.8	4.6	3.3	6.3	2.7	7.2	

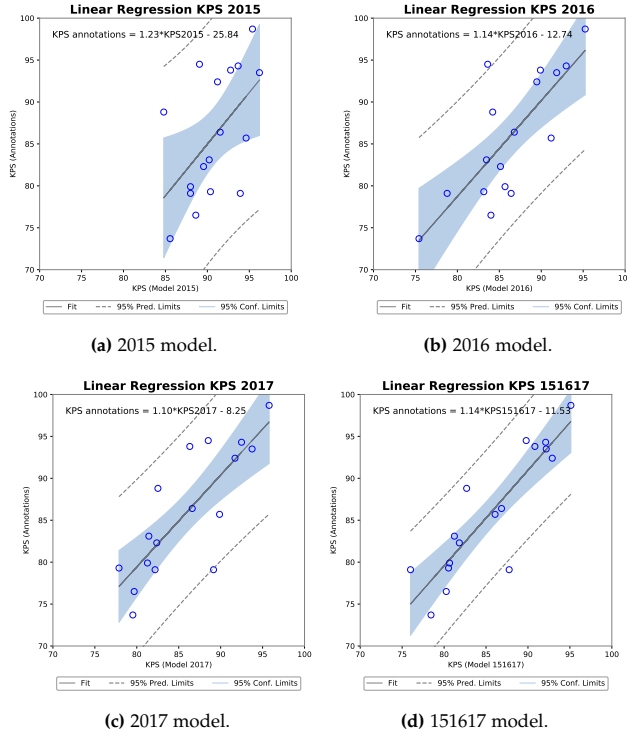
### 3.3 Correlation Analysis

Given the results in Table A.4 across the varying PGs, the effectiveness of the KPS calculation can be evaluated across a number of different potential sizes of kernel fragments. This section covers a correlation analysis for both the R-FCN and MNC method.

#### R-FCN

Four scatter plots including the equation describing the linear regression fit can be seen for each R-FCN model against the KPS annotations in Figure A.10. Each indicate a positive slope of an increasing KPS for a model as the ground truth KPS increases.

### 3. Results



**Fig. A.10:** Four scatter plots of the R-FCN model KPS against annotation KPS with linear regression analysis computer for each.

To determine the significance of a potential correlation, a Pearson's correlation coefficient was calculated as shown in Table A.5. Results from a Shapiro-Wilk normality test are also shown, as Pearson's assumes that both samples arise from a normal distribution. A high  $W$ , as present for all five samples in Table A.5, means that the null hypothesis that the population is normally distributed cannot be rejected. Following [34] we can interpret the results for Pearson's correlation coefficient that all models have a strong positive correlation to the annotation KPS. The strongest being the 151617 model of  $r(15) = 0.88$  with a p-value of 0.000003, explaining 77.7% of the variance in the ground truth KPS.

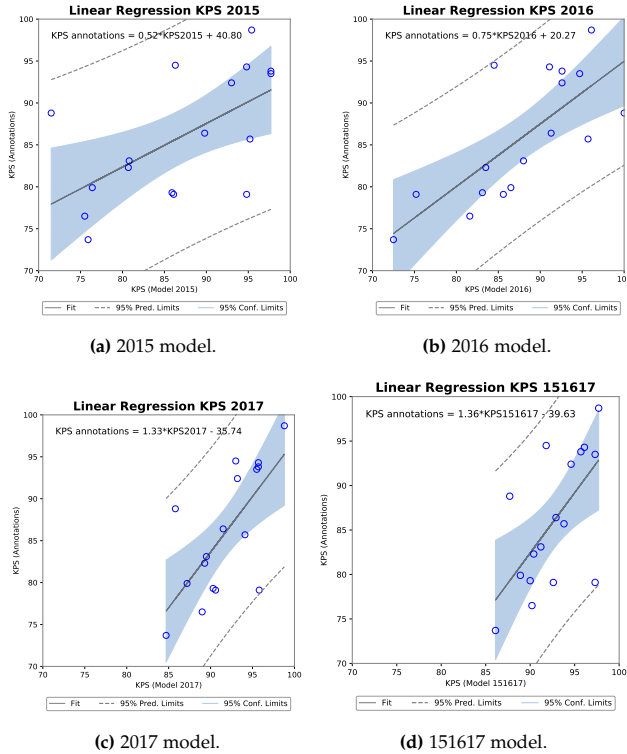
## Chapter A.

**Table A.5:** Correlation analysis via Pearson’s correlation coefficient for the KPS of the four R-FCN models against the annotation KPS. Pearson’s assumes a normal distribution in the data which is evaluated through a Shapiro-Wilk normality test.

KPS	Shapiro-Wilk		Pearson’s Correlation		
	W	p-value	r(15)	p-value	$r^2$ (%)
Annotations	0.94	0.32	NA	NA	NA
2015	0.973	0.870	0.54	0.0244	29.4
2016	0.97	0.816	0.77	0.0003	59.5
2017	0.94	0.320	0.81	0.00009	65.1
151617	0.94	0.327	0.88	0.000003	77.7

## MNC

The corresponding four scatter plots for the MNC models can be seen in Figure A.11. Again, a positive relationship is indicated between the KPS from each model and the KPS for annotations across processor gaps.



**Fig. A.11:** Four scatter plots of the MNC model KPS against annotation KPS with linear regression analysis computer for each.

## 4. Discussion

Table A.6 firstly show Shapiro-Wilk tests for each sample with high W and corresponding p-values. The resulting Pearson’s correlation coefficient also indicates a strong positive correlation. The strong appears from the 2016 model with  $r(15) = 0.74$  with a p-value of 0.0007, explaining 54.4% of the variance in the KPS annotations.

**Table A.6:** Correlation analysis via Pearson’s correlation coefficient for KPS of the four MNC models against the annotation KPS. Pearson’s assumes a normal distribution in the data which is evaluated through a Shapiro-Wilk normality test.

KPS	Shapiro-Wilk		Pearson’s Correlation		
	W	p-value	r(15)	p-value	$r^2$ (%)
Annotations	0.94	0.32			
2015	0.91	0.098	0.60	0.0106	36.2
2016	0.97	0.743	0.74	0.0007	54.4
2017	0.97	0.806	0.69	0.002	48.1
151617	0.97	0.666	0.63	0.0065	39.9

## 4 Discussion

The potential to train CNN models for kernel fragment recognition in RGB images of silage is promising. This appears to be the case even without conducting the time-consuming step of separating kernels and stover before evaluation, as in all current popular kernel fragmentation evaluation methods [3–6].

The four models trained in both R-FCN bounding-box and MNC instance segmentation performed well and two major tendencies appeared. Firstly and possibly unsurprisingly, a larger training dataset, such as that of 151617, led to models that performed well across all metrics on all test sets. Deep learning methods are known to have a high requirement on the amount of data and the roughly  $10\times$  larger 151617 training set in comparison to the 2015 and 2016 sets seemed to show this effect. However, a total of 1393 images with 6907 annotated kernel instances is not on the same level as considerably larger object recognition benchmarks such as PASCAL VOC [27] or MS COCO [28] consisting of over 10,000 and 165,000 images for training respectively. The trained R-FCN and MNC models of course take advantage of transfer learning from a pre-trained models on ImageNet datasets. With this aid, roughly 1400 annotated training images in 151617 set gave consistent results across test images from three different harvest years. Additionally, the second finding was of the at times significant improvement when adding only a small amount of data to a larger dataset. This was seen for the models trained on the 151617 dataset for test sets 2015 and 2016, where despite the

additional data being in the minority during training in contrast to images from 2017, they had a large increase in performance compared to models that did not combine all of the data.

With respect to the viability of using a CNN-based model for KPS measurement, both methods can be deemed to have potential. A strong positive correlation was found between annotation KPS and model KPS, with the strongest existing for the 151617 R-FCN model. A criticism of the correlation analysis is naturally that this was against annotation KPS and not a truer laboratory measurement than in [6]. However, as the training and testing splits were kept separate, the correlation results still give a good indication for the approaches.

In comparison to [6] who show KPS measurement given manually separated kernels in a controlled camera setting, the error measurement across sequences is similar to our work. KPS based on image analysis from wet samples from the field from [6] show an average absolute error of 5.6% in comparison to our range of 2.7% to 7.2% dependent on the model and test set. Of course, care should be taken comparing the two works given the differences in ground truth measurement, location of harvesting, the machine, and so forth. A key improvement in this work is the time required to obtain a KPS measurement. In [6] the time was improved to hours instead of days as in [3], however, due to removing the requirement of kernel/stover separation, this work allows KPS calculation to be done in minutes.

Future work is to evaluate against a laboratory measured KPS as mentioned earlier. Furthermore, research into applying newer object recognition methods from the fast-moving field may also be viable, potentially improving challenges such as recognition of small objects. Finally, such CNN-based methods could be used to measure other silage-quality aspects, such as the cutting length of the forage harvester.

## 5 Conclusions

This work has shown that kernel fragmentation in maize silage can be estimated from images using trained CNNs in both bounding-box and instance segmentation form. Through transfer learning and training models on images captured across three different harvest seasons, both forms were able to estimate the fragmentation robustly. This was evaluated via computer vision metrics and an analysis of the correlation between model predictions and a kernel processing score. Where the latter showed a strong correlation for both CNN forms to an industry standard kernel processing score.

Furthermore, this work showed promise in kernel fragmentation estimation in non-separated kernel/stover images, leading to a potentially significant decrease in measurement time.

## References

- [1] L. Johnson, J. Harrison, D. Davidson, W. Mahanna, and K. Shinnors, "Corn silage management: Effects of hybrid, chop length, and mechanical processing on digestion and energy content," *Journal of dairy science*, vol. 86, pp. 208–31, 02 2003.
- [2] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [3] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," In: *Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [4] J. Heinrichs and M. J. Coleen, "Penn state particle separator," May 2013. [Online]. Available: <https://extension.psu.edu/penn-state-particle-separator>(accessedon24July2018)
- [5] "Making sure your kernel processor is doing its job." *Focus Forage*, vol. 15, pp. 1–3, 2014.
- [6] J. L. Drewry, B. D. Luck, R. M. Willett, E. M. Rocha, and J. D. Harmon, "Predicting kernel processing score of harvested and processed corn silage via image processing techniques," *Computers and Electronics in Agriculture*, vol. 160, pp. 144 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918311955>
- [7] H. Kaur and B. Singh, "Classification and grading rice using multi-class svm," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, pp. 1–5, 2013.
- [8] A. Aggarwal and R. Mohan, "Aspect ratio analysis using image processing for rice grain quality," *International Journal of Food Engineering*, vol. 6, 01 2010.
- [9] F. Antonucci, S. Figorilli, C. Costa, F. Pallottino, A. Spanu, and P. Mene-satti, "An open source conveyor belt prototype for image analysis-based rice yield determination," *Food and Bioprocess Technology*, vol. 10, pp. 1–8, 02 2017.
- [10] G. Dalen, "Determination of the size distribution and percentage of broken kernels of rice using flatbed scanning and image analysis," *Food Research International*, vol. 37, pp. 51–58, 06 2004.

## References

- [11] P. Duboscclard, S. Larnier, H. Konik, A. Herbulot, and M. Devy, "Automatic visual grading of grain products by machine vision," *Journal of Electronic Imaging*, vol. 24, p. 061116, 11 2015.
- [12] N. Visen, J. Paliwal, D. Jayas, and N. White, "Image analysis of bulk grain samples using neural networks," *Canadian Biosystems Engineering / Le Genie des biosystems au Canada*, vol. 46, 01 2003.
- [13] B. Anami and D. Savakar, "Effect of foreign bodies on recognition and classification of bulk food grains image samples," *Journal of Applied Computer Science & Mathematics*, vol. 3, 01 2009.
- [14] C. Lee, L. Yan, T. Wang, S. Lee, and C. Park, "Intelligent classification methods of grain kernels using computer vision analysis," *Measurement Science and Technology*, vol. 22, p. 064006, 05 2011.
- [15] F. Guevara-Hernandez and J. Gomez-Gil, "A machine vision system for classification of wheat and barley grain kernels," *Spanish Journal of Agricultural Research*, vol. 9, p. 672, 09 2011.
- [16] N. Patil, V. Malemath, and R. M. Yadahalli, "Color and texture based identification and classification of food grains using different color models and haralick features," *International Journal on Computer Science and Engineering*, vol. 3, 12 2011.
- [17] A. Miao, J. Zhuang, Y. Tang, L. He, X. Chu, and S. Luo, "Hyperspectral image-based variety classification of waxy maize seeds by the t-sne model and procrustes analysis," *Sensors*, vol. 18, p. 4391, 12 2018.
- [18] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.
- [19] S. Skovsen, M. Dyrmann, A. Krogh Mortensen, K. Steen, O. Green, J. Eriksen, R. Gislum, R. Jørgensen, and H. Karstoft, "Estimation of the botanical composition of clover-grass leys from rgb images using data simulation and fully convolutional neural networks," *Sensors*, vol. 17, p. 2930, 12 2017.
- [20] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, p. 2022, 09 2017.
- [21] D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 797–804.



## References

- [22] S. Mohanty, D. Hughes, and M. Salathe, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 04 2016.
- [23] A. Milioto, P. Lottes, and C. Stachniss, "Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W3, pp. 41–48, 08 2017.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [25] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," 12 2016, pp. 379–387.
- [26] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," 06 2016, pp. 3150–3158.
- [27] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010.
- [28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, "Microsoft coco: Common objects in context," vol. 8693, 04 2014.
- [29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 06 2016, pp. 770–778.
- [32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 1–42, 01 2015.
- [33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, 06 2014.

## References

- [34] J. Cohen, "Statistical power analysis for the behavioral sciences," *SER-BIULA (sistema Librum 2.0)*, vol. 2nd, 01 1988.

# Paper B

## Anchor Tuning in Faster R-CNN for Measuring Corn Silage Physical Characteristics

Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B.  
Moeslund

The paper has been published in  
*Elsevier Computers and Electronics in Agriculture* Vol. 188, pp. 106344, 2021.

Please note that at the end of this paper we have added an appendix covering experiments for a number of design choices for the networks used in this work. This is done to provide more insights for the interested reader.

© 2021 Elsevier  
*The layout has been revised.*

### Abstract

*Efficient measurement of harvested corn silage from forage harvesters can be a critical tool for a farmer. Suboptimal fragmentation of kernels can affect milk yield from dairy cows when the silage is used as fodder and oversized stover particles can promote mould yielding bacteria during storage due to resulting air pockets. As a forage harvester can harvest hundreds of tonnes per hour, an efficient and robust system for measuring quality in the field is required, however, current methods require manual errorsome separation steps or for samples to be sent to an off-site laboratory. Therefore, we propose to adopt Faster R-CNN with an Inceptionv2 backbone to detect kernel fragments and oversized particles in images of corn silage taken directly after harvesting without the need for separating particles. We explore strategies of data sampling for specialist models, transfer learning from differing domains and tuning the anchors in the Region Proposal Network to accommodate for changes in object shapes and sizes. Our approach leads to significant improvements in average precision for kernel fragmentation and stover overlengths of up to 45.2% compared to a naive model development approach, despite the challenging cluttered scenes. Additionally, our models are able to predict quality for network predictions with the Corn Silage Processing Score (CSPS) for kernel fragmentation and a measure we introduce for chopped stover named Overlength Processing Score (OVPS). For both scores we obtain a strong correlation against physically measured samples with an  $r^2$  of 0.66 for CSPS, 0.79 and 0.95 for OVPS at two verbal theoretical length of cut.*

### 1 Introduction

The evaluation of quality of harvested corn silage is a critical step for a farmer. The farmer has two key settings to adjust during corn silage harvesting with a forage harvester. Firstly, the Processor Gap (PG), where two mill rolls compress and crack open kernels into fragments with a gap of a few millimetres. Secondly, the Theoretical Length of Cut (TLOC) which controls the cut of the stover (leaves and stalks of the plant) to a desired length by a rotating drum consisting of a number of knives. Ensuring the corn silage is harvested to the appropriate size is one of the most important aspects when a farmer harvests. Corn kernels should be cracked open such that the starch content can easily be accessed when being fed for dairy cows and the stover should be cut such that it encourages saliva production through cud chewing such that a proper rumen pH is maintained in the cow [1]. The stover should also be chopped such that it allows for compact packing and storage during fermentation in the silo. Longer pieces of stover can create pockets of air between the particles allowing for aerobic bacteria to grow which in turn can produce mould and yeast ruining the silage [1]. A forage harvester is able to harvest massive amounts of silage in a short period of time and therefore suboptimal ma-

chine settings can greatly affect both the fuel consumption and the resulting yield. Depending on the machine settings, modern forage harvesters use between 130 to 180 litres of fuel and can harvest between 200 to 300 tonnes per hour [2]. Typically the farmer selects the machine settings based upon their expertise and their given field conditions. However, within a field there can be considerable variations in the corn plant dependent on numerous factors such as moisture level and differences in plant maturity requiring adjustments to the PG and TLOC. Naturally this places a large requirement on the operator and automating this process would place lower requirements on the farmer while aiding in optimising both yield and machine usage.

Industry standards for determining silage quality is through manual measurements of the particle size distribution. For a faster measurement in the field a farmer can utilise the Penn State Particle Separator (PSPS) [1], where the farmer shakes three or four stacked trays each consisting of a specific sieve gap such that the sample can be separated and subsequently weighed. Laboratory measurements can also be conducted which require a sample of corn silage to be sent off-site for mechanical sieving in order to gain a more precise measurement such as with the ASABE particle separator [3]. The mechanical separation removes the potential human error as could occur with the PSPS but naturally is much more time-consuming and does not allow the farmer to gain insight into the harvester settings during harvesting. If only the kernel fragmentation is of interest, a farmer can measure the Corn Silage Processing Score (CSPS) [4] of kernels separated from the stover using a Ro-Tap sieving system, here, the percentage of particles that pass a 4.75 mm sieve defines the processing quality. For the stover portions of the silage the aim is often to measure particle lengths in order to promote physically effective Neutral Detergent Fibre which increases chewing and healthy rumen pH [5]. A common metric is to measure the mean particle length of a stover sample. Previous work has been conducted on the measurement of the fragmentation level and estimating CSPS through computer vision methods [6–9]. However, minimal literature exists on the measurement of chopped stover and those previous works require manual separation such that there is no overlap between particles [10, 11]. In this work we tackle the much harder problem of a fully automated approach. To this end, we in this work show the effectiveness of modern deep learning architectures for both measuring kernel and stover particles. We adopt the same approach for estimating CSPS for kernel fragments as in [8] where the aim is to localise and measure all kernel fragments in the image. However, if we were also to follow industry standards for stover measurement, such as to estimate mean particle size or particle size distribution, detection of all relevant instances in an entire sample would be required. Such a task is not feasible for object recognition systems with the high levels of clutter and occlusion that occur in non-separated samples. This challenging difference is shown in Figure B.1



**Fig. B.1:** Example of harvested silage. The white outline shows the kernel fragments and all remaining particles in the image are stover.

where kernel fragments are outlined in white and the remaining particles in the image are stover.

Therefore, we propose to introduce measuring only large instances of stover as overlenghts and thereby addressing key quality aspects directly relating to feed quality and factors that can lead to spoiled silage during storage. We show an approach to estimate the portion of stover overlenghts in samples for two different verbal TLOCs, namely, 4 and 12 mm. Chopping strategies for corn silage can differ depending on the farm based on feeding and storage. Therefore, we show results for two different verbal TLOCs rather than having a single metric. Our definition of an overlenght is  $1.5 \times$  verbal TLOC and in this work we create three datasets with different verbal TLOCs in order to evaluate this premise across multiple stover lengths. We believe that the compromise of only estimating overlenghts in the cluttered non-separated samples can give a strong indicators to a farmer directly in the field which can complement other metrics such as mean particle length that require additional manual steps. Further explanation of the datasets are covered in Section 3.1 together with the annotation process. The varying overlenght definition places different requirements on a system as it should be able to adjust to changes in verbal TLOC. We therefore explore strategies for model training in data separation for the development of specialist models for a given verbal TLOC and adjusting parameters of the network to tune towards specific overlenght object sizes. Our exploration leads us to show that precision and quality measurement can be significantly improved for both overlenght and kernel recognition compared to a naive model development approach. Models are evaluated two-fold, with object recognition metrics against manually annotated instances and against images with accompanying physically measured quality scores. By testing our models with

two approaches we are able to determine model optimisation trends that improve our results but also see that due to the challenging cluttered scenes it is important to include a method completely separate of human bias in annotation.

Our contribution is three-fold:

- We show for the first time how overlengths can be analysed automatically from images without the need for separation of particles and hence pave the way for a system that can efficiently aid the farmer in adjusting machine settings of their forage harvester without errorsome or time-consuming sieving methods.
- We adopt a two-stage recognition network for the tasks of kernel and overlength recognition, namely Faster R-CNN [12] with an Inceptionv2 [13] backbone, showing the robustness of the system despite low number of annotated instances in scenes with high amounts of clutter and occlusion.
- We show through strategies for data sampling, transfer learning and tuning of parameters of the Region Proposal Network (RPN) significant improvements in Average Precision (AP) and correlation against physical measurements compared to a naive training approach.

## 2 Related Work

Recognition and localisation of objects for quality control is a key area of research in computer vision. In agriculture, harvest inspection has been explored with both classical approaches such as feature extraction in combination with a trained classifier [14–19] or through deep learning systems [20–22]. Within corn silage there is limited work for measuring the quality using computer vision. For stover measurement there are a few that provide a particle size distribution of the entire sample using classical computer vision and determine geometric characteristics of the particles [10, 11], however, both require all particles to be separated and placed in a controlled setting. Within kernel fragmentation, firstly, [6] determine the maximum inscribed circle within fragments through classical computer vision approaches of kernel samples separated from the stover and spread out on a black background and in [7] the same approach is used to determine the fragmentation of in situ disappearing dry matter. In [8, 9], the authors measure the fragmentation in non-separated samples taken directly from the harvester using two-stage recognition Convolutional Neural Networks (CNNs) to predict instance segmentation and bounding-boxes, however, CSPS predicted from the networks is evaluated against an estimated CSPS from annotations.



## 2. Related Work

The determination of particles sizes in machine vision is present in both agriculture but also in other industries. Inspection of minerals is one such domain, where classical approaches to determine shape and size characteristics have been extracted and resulting measurements compared to mechanical sieving distributions [23–25]. Deep learning approaches through CNNs have also been adopted for the task, such as in [26] where a Mask R-CNN was trained to predict the location and classes of agglomerate nanoparticles from which size information could be extracted. In [27] U-Net was adopted for droplet size distribution in chemical engineering applications. In [28] a custom CNN has proposed to directly predict the histogram of object sizes of images containing fly larvae.

In modern object detection, usage of region proposals is common practice through the RPN since it was presented in Faster R-CNN [12]. In the RPN, predefined priors, known as anchors shapes, are used to densely predict object proposals at sliding window locations in the computed feature map. The anchors shapes were densely set at three scales ( $128^2$ ,  $256^2$ ,  $512^2$ ) and three aspect ratios (1:1, 1:2, 2:1) to cover a variety of potential shapes and sizes. While the RPN provided significant improvements and is still a robust module in an object detection pipeline a number of methods have attempted to improve the dense anchoring scheme. In YOLOv2 [29], the RPN was adopted as it was found that the original YOLO made errors in terms of localisation and recall. However, rather than having hand-picked anchor priors, the boxes were determined using k-means clustering with an Intersection-over-Union (IoU) distance metric between cluster centroids and annotated training bounding-boxes. In [30] guided anchoring was introduced in the RPN to use semantic features to learn the location and shapes of the anchors into each level of the feature pyramid network. RefineDet [31] used an anchor refinement module that filters negative anchors such that the classification step is simplified and adjusts anchors over a cascade of decreasing feature maps. In MetaAnchor [32] meta-learning is used in anchor generation that allows the anchor box priors to be set at inference time rather than during training.

A number of examples exist of adjusting anchor boxes for specific applications resulting in better performance. Firstly, for pedestrian detection smaller scale priors in [33] and the specific aspect ratio of 0.41 in [34]. For face detection in [35] an aspect ratio of 1:1 was used as faces are generally square in shape. For text detection in [36] a number of higher and wider aspect ratios were adopted. For ship detection, [37] used hand-picked rotations and scales in an encoder-decoder network.

### 3 Methodology

This section covers the datasets used in this work for localising kernel fragments and stover overlengths, both for training and evaluating models with hand labelled annotations and for validating models against physically measured samples with relevant corn silage physical characteristics metrics. We also give an overview of our methods for improving our models by training specialist models on subsets of data, transfer learning, adding a post-processing filtering step and tuning anchors in the RPN for our specific tasks. Finally, we cover how we converted predictions to corn silage metrics in order to compare against physical samples.

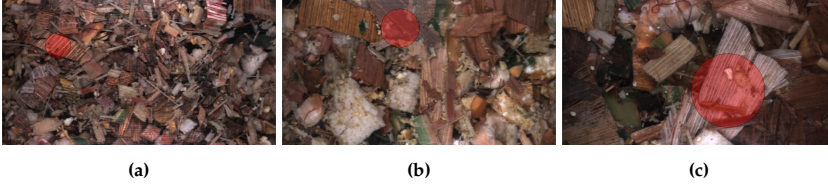
#### 3.1 Data Collection and Annotation

In this work we used separate annotated datasets for kernel fragmentation and overlength measurement. We adopted the same kernel fragmentation dataset as in [8, 9] consisting of 11601 kernel fragment instances annotated in 2438 RGB images. The images were captured across three separate harvesting seasons in 2015, 2016 and 2017.

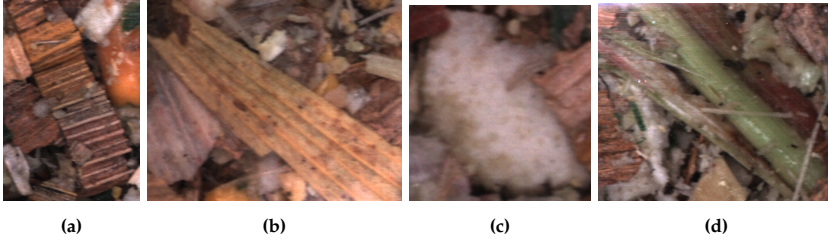
For overlength recognition we created a new dataset of RGB images of the non-separated corn silage after harvesting with three different verbal TLOCs producing separate size distributions of chopped stover. While it could have been possible to extend the kernel dataset used in [8] to include overlengths the decision was made to create a separate dataset. The original dataset did not have variation across a large enough number of cutting lengths and would not allow to evaluate the system over changing conditions as the machine settings are altered. Images were taken directly after harvesting with a constant distance between the camera and silage sample allowing for a conversion between pixels and millimetres such that a quality score could be estimated for instances. From these images, four datasets were defined: *Small* containing images of silage harvested with a 4 mm verbal TLOC, *Medium* harvested at 6 mm verbal TLOC, *Large* harvested at 11.5 mm verbal TLOC, and finally *All* which is a combination of the images from the previous three datasets. Figure B.2 shows an example image from each of the three size datasets including a red circle which diameter shows the overlength definition for the given verbal TLOC shown during annotation.

Figure B.3 shows the four class definitions used in this work when differentiating between overlengths. Firstly, we show the classes accepted leaves in Figure B.3 (a) and non-accepted leaves in Figure B.3 (b) where in both cases the leaves extend beyond the overlength indicator but there is a difference in the classes based on the structure of the plant. For non-accepted leaves the longest axis of the plant follows the fibre structure but for accepted leaves the axis that exceeds does not. The definition between these types of leaves

### 3. Methodology



**Fig. B.2:** Example of images from the datasets of different TLOCs. (a) example from *Small* (4 mm). (b) example from *Medium* (6 mm). (c) example from *Large* (11.5 mm). For each image the diameter of the red circle indicates the overlenth definition of  $1.5 \times \text{TLOC}$ .



**Fig. B.3:** An example from each of the four classes from silage harvested with an TLOC of 4 mm. (a) accepted leaves, (b) non-accepted leaves, (c) inner stalk, (d) outer stalk.

is due to the manner in which the plant is fed into the harvester. The header of the harvester cuts near the bottom of the corn plant and feeds it from this end into the machine where the rotating drum chops perpendicularly. As the drum only chops across across one axis it is considered a critical error when the leaf is too long in the axis following the fibres but not along the other. The example in B.3 (a) can occur as the leaf is wrapped around the plant and unravels after passing through the cutting drum. This is difficult to address from a machine stand-point as these types of leaves are considered "accepted" as the machine is cutting stover in the manner it is designed to cut. In Figure B.3 (c) we show the class inner-stalk and in Figure B.3 (d) outer stalk. As the naming denotes, these are two separate parts of the stalk plant and are different in terms of digestion for the dairy cows and how the compress during storage.

Images were annotated producing bounding-boxes for each instance. Table B.1 shows an overview of the datasets for the three TLOCs. Firstly, it can be seen that the number of instances per image is greater for a smaller TLOC. As a forage harvester can output hundreds of tonnes per hour this puts a larger requirement on the the rotating cutting drum at 4 mm compared to 11.5 mm. Additionally, due to the definition there is a more stringent threshold in relation to the machine setting as a 4 mm TLOC overlenth is at 6 mm compared to a 11.5 mm TLOC at 17.25 mm.

**Table B.1:** Annotation statistics for instances for the three verbal TLOCs. A decreasing number of instances occur for a larger verbal TLOC but with an increasing bounding-box size (pixels).

TLOC	Images	Instances	Accepted leaves	Non-accepted leaves	Inner stalk	Outer stalk	Avg. size	Avg. major axis length	Avg. minor axis length
4	163	1233	520	419	75	209	14518.9	216.6	94.3
6	199	904	182	559	35	122	26315	294.3	122.7
11.5	113	263	51	172	1	38	61328.2	485.5	179.9

### Physical Samples

In addition to the annotated datasets for both kernel fragmentation and overlengths we used a third dataset for validation of the models. Multiple image sets were captured across two harvested weeks (CW40 & CW43) with varying machine settings. A total of 10 image sets were captured at verbal TLOC 4 mm and 15 at 12 mm. In addition to the two different verbal TLOCs the kernel processor was varied between image sets with roll gaps of either 1, 2 or 3 mm. For each image set a sample of corn silage was taken and physically measured for both kernel fragmentation and overlengths. Kernel fragmentation was measured using CSPS [4] by sieving a 600g sample and determining the percentage of particles passing 4.75 mm. The overlengths were measured using 20-30 kg samples where the percentage passing a sieve corresponding to  $1.5 \times$  verbal TLOC gave the distribution. More specifically, samples harvested at verbal TLOC 4 mm were measured against a 6 mm sieve compared to an 18 mm sieve at verbal TLOC 12 mm.

## 3.2 Model Training

Faster R-CNN variants for both kernel and overlengths were trained with a number of common parameters using the TensorFlow object detection API [38] with TensorFlow version 1.13.1 on an NVIDIA Titan XP GPU. Images were cropped such that only the silage could be seen in the frame and resized to  $600 \times 1200$  during training and testing. Each model variant was trained for a total of 25,000 iterations and the iteration with the lowest validation loss was chosen for testing. Each of the datasets were split into 70% training, 15% validation and 15% testing. It is important to note that the *All* dataset for overlengths comprises the same data as the respective sets for each specific verbal TLOC such that the test results are comparable.

Models were optimised using stochastic gradient descent with a learning rate of 0.002, momentum of 0.9 and a batch size of 1. A maximum of 300 proposals were sampled per image in the RPN at an IoU threshold of 0.7 for positive examples and IoU threshold below 0.3 for background with an annotated instance. Overlapping detections from the network are removed with an IoU threshold of 0.6 with non-maximum suppression.

Transfer learning is conducted for each of the models where weights are

initialised from a model trained on COCO [39] available from the TensorFlow object detection API. In the case of overlengths we additionally finetune towards a specific verbal TLOC from a model trained on the *All* dataset initially finetuned from COCO.

#### 3.3 Filtering Predictions

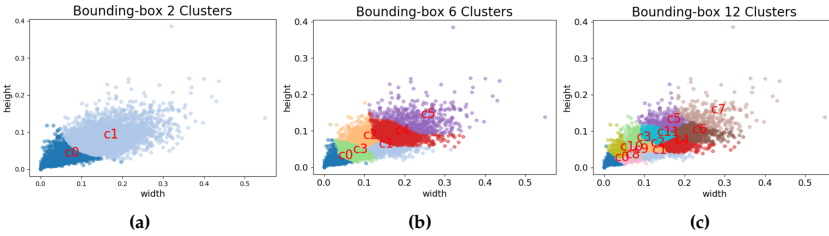
In the case of overlengths, specialist models are trained on a subset of the data and thereby have the overlength definition information indirectly given to the models through the annotations. However, for models trained on the *All* dataset this is not given and predictions for overlengths may be made towards smaller objects despite the verbal TLOC being set at a larger length. As the verbal TLOC is given by a farmer at inference time we also evaluate incorporating this into models by removing predictions below the appropriate overlength definition threshold in a post-processing step. For example, performing inference on the *Large* test set we filter any predictions where the major axis is less than  $1.5 \times 11.5$  mm.

#### 3.4 Anchor Clustering

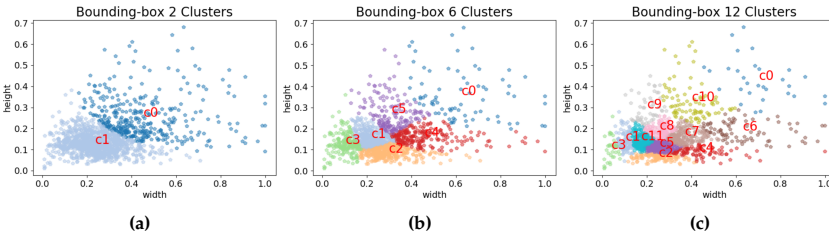
The RPN module in Faster R-CNN has the task of finding a number of object proposals that are likely to contain an object. The RPN is a light-weight module and allows for more complexity to be used in the feature extraction, classification and localisation stages. A number of parameters exist that can aid in proposal generation including the anchor shapes and sizes. For general object detection, such as in benchmark challenges COCO [39] or PASCAL VOC [40] where object shapes can vary greatly in both shape and size, anchors boxes are densely set to cover many scales of both square and rectangular boxes. However, if a dataset is more specialised the anchor boxes can be specified accordingly aiding anchor refinement training in the RPN when priors are closer to the "true" shape.

To determine the shape and sizes of the anchors for a model we sample all of the annotated bounding-boxes for a given dataset. An example is shown in Figures B.4 and B.5 where each of the three subfigures have the normalised widths and heights of each bounding-box for the kernel and overlength *All* training dataset respectively. Cluster centres defining an anchor shape are found using k-means with the distance metric between the IoU of centroids and annotations as in [29]. We see in Figure B.4 that the annotations for kernels have a trend to be slightly longer along their width and sizes largely between 0.1 to 0.2 along both axes. Figure B.5 show overlengths have a greater variation in sizes due to the differing overlength definition. In addition, we also see that the differences in the height and widths for an instances are considerably larger than kernel annotations.

Typically for larger datasets a larger number of anchors are used, for example, the provided Faster R-CNN models in the TensorFlow object detection API [38] have 12 anchors. However, for our specific case the optimal number is not known, therefore, we experiment with the number of anchor boxes for a model, as shown in Figures B.4 and B.5, where we calculate anchor cluster centres for two anchors (a), six anchors (b), and 12 anchors (c). When clustering for a specific verbal TLOC we take the relevant subset of bounding-boxes in Figure B.5 and tune anchors size accordingly further refining the anchor priors and potentially narrow the requirements on RPN optimisation.



**Fig. B.4:** Anchor height and widths found using k-means clustering for the kernel training set for (a) 2 anchors, (b) 6 anchors and (c) 12 anchors. The anchor centroid determined with k-means is shown by the red text. Each point is the normalised height and width for a bounding-box annotation.



**Fig. B.5:** Anchor height and widths found using k-means clustering for the overlength *All* training set for (a) 2 anchors, (b) 6 anchors and (c) 12 anchors. The anchor centroid determined with k-means is shown by the red text. Each point is the normalised height and width for a bounding-box annotation.

### 3.5 Converting Predictions to Silage Quality Measurement

Faster R-CNN outputs bounding-box coordinates together with a confidence score for each prediction. In order to determine the quality of harvested silage from a set of images predictions must be converted to a quality metric, here, we adopt the industry standard CSPS for kernel fragmentation and introduce the metric Overlength Particle Score (OVPS) for stover. As the distance between camera and corn silage samples is constant we can convert the

### 3. Methodology

number of pixels to millimetres for a prediction and compute the quality in images. For each bounding-box prediction we determine the major axis of the box and if the length is below 4.75 mm (95 pixels) the instance is considered a correctly fragmented kernel. We estimated CSPS in two ways, by computing the percentage below the threshold by the number of instances against all instance predictions and by the number of pixels within the bounding-boxes against all bounding-box pixels. Taking the percentage of correctly fragmented kernels above a confidence threshold in relation to all kernel predictions gives the estimated CSPS. Other options exist for converting predictions to the quality scores such as using the minor axis or the radius of the maximum inscribed circle [6]. In our case initial investigations showed scores that correlated well and were closer to the true score by thresholding the major axis.

In the case of OVPS we aim to predict the percentage of overlengths in a sample as an estimation of weight. For the total number of overlengths in an image set we compute the percentage of pixels accounting for overlength bounding-boxes over the total number of pixels in the image set. This allows us to estimate the weight of the overlengths which can be compared to the sieving measurements.

#### 3.6 Computer Vision Analysis

Predictions from the deep learning models are firstly evaluated using the COCO [39] object detection metric denoted AP which is averaged over 10 IoU thresholds between 0.5 to 0.95 with steps of 0.05. Additionally, we analysed the predictions using the PASCAL VOC [40] approach of AP at IoU 0.5 (AP@0.5) and a stricter IoU metrics at 0.75 (AP@0.75).

#### 3.7 Statistical Analysis

For the 25 image sets the correlation was evaluated for CSPS and OVPS between the model and physical samples using Pearson's Correlation Coefficient (PCC) and the  $r^2$  coefficient of determination. Furthermore, the Root Mean Square Error (RMSE) between the model scores and physical scores were calculated. To measure agreement we used Lin's Concordance Correlation Coefficient (CCC) to show the capability of our model measurements against the gold standard of physical measurements. In addition to CCC, we also show relevant Bland-Altman plots further assessing agreement.

## 4 Results

This section describes the results for kernel fragmentation and overlenth recognition using the Faster R-CNN variants. We compare our model variants against a baseline naive Faster R-CNN trained with standard parameters define in Section 3.2. This way we can evaluate and compare against a standard practice of deep learning development of simply training on a large dataset. The naive models are denoted *Baseline* for kernel fragmentation and *All* for overlenthgs.

### 4.1 Kernel Fragmentation

For kernel fragmentation we aim to extend our models in comparison to [8, 9] by adding anchor tuning to our two-stage networks. In [9] a Faster R-CNN with Inceptionv2 was trained using a naive strategy with parameters provided from TensorFlow API [38] for kernel fragmentation on the same dataset as this work. We name this model baseline in order to directly show the effect of our extensions. Table B.2 shows the AP results on the test set for kernel fragmentation. We see significant improvements using anchor tuning in all three cases against a naive baseline Faster R-CNN. The number of anchors tuned is shown by  $xa$  where  $x$  is the number of anchors. The model with two anchors tuned for the task provides the best results and increases all metrics by a number of percentage points (pp).

**Table B.2:** Faster R-CNN results on the kernel 151617 test set. Results are shown with against a baseline naive training strategy and with tuning for either 2, 6, 12 anchors.

Model	AP	AP@0.5	AP@0.75
R-FCN [8]	N/A	34.0	NA
MNC [8]	N/A	36.1	NA
Baseline [9]	25.6	51.9	22.3
2a	<b>28.5</b>	<b>56.6</b>	<b>25.7</b>
6a	27.4	55.9	24.0
12a	26.0	54.0	21.0

Table B.3 shows the correlation scores between predicted CSPS based on instances counts below the CSPS threshold and Table B.4 for CSPS computed with pixels below against physical CSPS over the two harvest weeks for predictions above 50% confidence. Both methods show improvements compared to their respective baselines in terms of PCC and  $r^2$ . However, in CW43 using CSPS estimated with instance counts in Table B.3 the baseline method performs best. Slight improvements are seen for the week in Table B.5 for 2a



#### 4. Results

and 6a. Overall when combining measurements from both harvest weeks the 2a model with CSPS estimated based on pixels has the highest correlation.

**Table B.3:** CSPS estimated with instance counts correlation between model estimation and physically measured samples across two harvest weeks.

Model	CW40				CW43				CW40+CW43			
	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
Baseline [9]	0.68	0.46	8.12	0.62	<b>0.64</b>	<b>0.41</b>	17.09	0.29	0.53	0.28	14.2	0.34
2a	<b>0.84</b>	<b>0.70</b>	5.39	0.80	0.63	0.40	8.89	0.54	<b>0.64</b>	<b>0.40</b>	7.69	0.62
6a	0.83	0.69	8.14	0.67	0.61	0.37	20.13	0.21	0.54	0.29	16.42	0.28
12a	0.64	0.47	11.05	0.51	0.52	0.27	17.74	0.23	0.52	0.28	15.42	0.29

**Table B.4:** CSPS estimated with pixel distribution correlation between model estimation and physically measured samples across two harvest weeks.

Model	CW40				CW43				CW40+CW43			
	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
Baseline [9]	0.73	0.53	38.71	0.07	0.66	0.44	44.76	0.05	0.64	0.41	42.44	0.06
2a	<b>0.79</b>	<b>0.62</b>	29.59	0.14	0.70	0.48	36.86	0.07	<b>0.66</b>	<b>0.43</b>	42.44	0.06
6a	0.76	0.58	39.19	0.07	<b>0.71</b>	<b>0.50</b>	46.72	0.07	0.64	0.41	43.86	0.05
12a	0.66	0.44	38.85	0.07	0.62	0.38	44.19	0.04	0.59	0.35	42.17	0.05

However, we observed that the Faster R-CNN models lacked small predictions and considerable improvements in PCC and r<sup>2</sup> were made when lowering the confidence threshold. This can be seen in Tables B.5 and B.6 with an optimal threshold for the 2a model appearing around 0.005 to 0.05. However, RMSE and CCC drop significantly showing that despite strong correlation a potential system with this model should be compensated appropriately. In both harvest weeks we see improvements and strong correlation to physical measurements. When decreasing the confidence threshold to this level the correlation scores are slightly better for the pixel-based CSPS, however, combining the two weeks show the same scores with 0.81 and 0.66 for PCC and r<sup>2</sup> respectively. This final best performing model, 2a at confidence threshold of 0.05, for PCC has an associated p-value to PCC of 2.25e-5, 0.0004 and 1.04e-6 for CW40, CW43 and CW40+CW43.

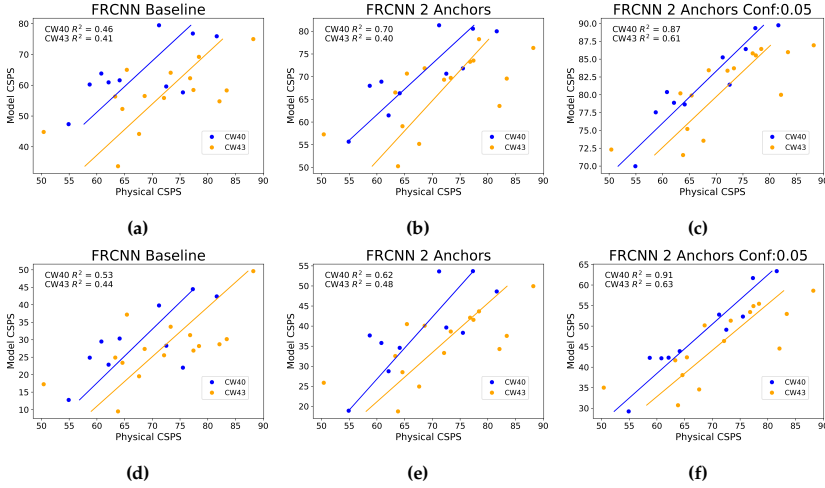
**Table B.5:** CSPS estimated with instance counts correlation at different confidence thresholds from the Faster R-CNN 2a model.

Model	CW40				CW43				CW40+CW43			
	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
Baseline [9]	0.68	0.46	8.12	0.62	0.64	0.41	17.09	0.29	0.53	0.28	14.2	0.34
0.5	0.84	0.70	<b>5.39</b>	<b>0.80</b>	0.63	0.40	8.89	0.54	0.64	0.40	<b>7.69</b>	<b>0.62</b>
0.25	0.90	0.80	8.90	0.57	0.66	0.44	<b>7.64</b>	0.57	0.71	0.51	8.17	0.57
0.05	<b>0.94</b>	<b>0.87</b>	25.35	0.19	<b>0.78</b>	<b>0.61</b>	31.3	<b>0.10</b>	0.80	0.65	29.07	0.13
0.005	0.91	0.84	18.87	0.15	0.77	0.59	16.23	0.17	<b>0.81</b>	<b>0.66</b>	17.34	0.16
0.0005	0.86	0.75	21.35	0.10	0.71	0.50	18.85	0.10	0.77	0.59	19.89	0.10

In Figure B.6 we show CSPS estimated with both approaches, first, with instance counts for the baseline scatter plot in B.6(a), Faster R-CNN with 2

**Table B.6:** CSPS estimated with pixel distribution correlation at different confidence thresholds from the Faster R-CNN 2a model.

	CW40				CW43				CW40+CW43			
Model	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
Baseline [9]	0.73	0.53	38.71	0.07	0.66	0.44	44.76	0.05	0.64	0.41	42.44	0.06
0.5	0.79	0.62	29.59	0.14	0.70	0.48	36.86	0.07	0.66	0.43	42.44	0.06
0.25	0.88	0.77	25.35	0.19	0.71	0.50	31.30	0.10	0.73	0.53	29.07	0.13
0.05	<b>0.95</b>	<b>0.91</b>	20.17	0.28	<b>0.79</b>	<b>0.63</b>	26.32	0.15	<b>0.81</b>	0.65	24.05	0.19
0.005	0.92	0.85	16.72	<b>0.32</b>	0.77	0.60	20.85	0.20	<b>0.81</b>	<b>0.66</b>	19.30	0.24
0.0005	0.84	0.71	<b>16.65</b>	0.30	0.71	0.51	<b>18.40</b>	<b>0.23</b>	0.77	0.59	<b>17.72</b>	<b>0.26</b>

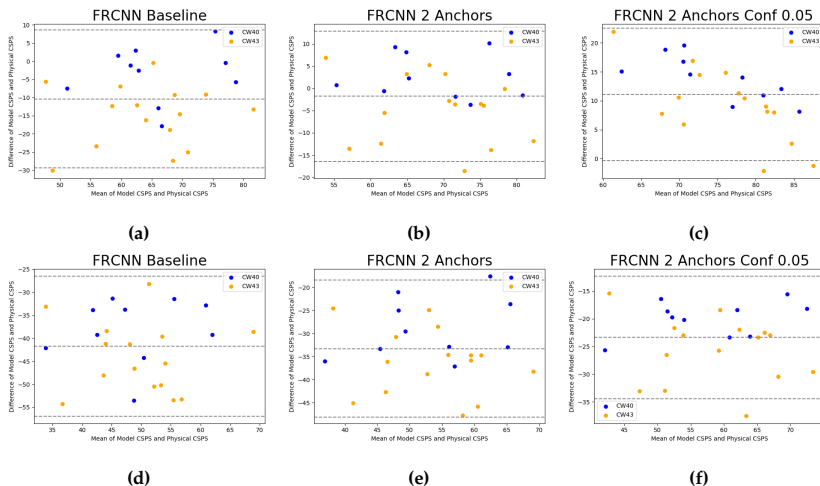
**Fig. B.6:** CSPS estimated with instance counts correlation for the baseline Faster R-CNN (a), Faster R-CNN 2a (b) and Faster R-CNN 2a with a confidence threshold of 0.05 (c). CSPS estimated with pixel distribution for baseline Faster R-CNN (d), Faster R-CNN 2a (e) and Faster R-CNN 2a with a confidence threshold of 0.05 (f).

anchors in B.6(b) and Faster R-CNN with 2 anchors at a confidence threshold of 0.05 in B.6(c). Secondly, for pixel bounding-boxes for baseline scatter plot in B.6(d), Faster R-CNN with 2 anchors in B.6(e) and Faster R-CNN with 2 anchors at a confidence threshold of 0.05 in B.6(f). In Figures B.6(c) and B.6(f) the strong correlation can be seen for the two weeks however concordance between points is also lower compared to counterpart models.

Finally, we show corresponding Bland-Altman plots for the models in Figure B.6 in Figure B.7 highlighting that 2a models at a confidence of 0.05 show a larger agreement by clustering closer together in terms of their difference, however, show a significantly larger difference to physical CSPS. compared to a more standard confidence threshold of 0.5.

Table B.7 and Figure B.8 show the effect of lowering the confidence threshold from 0.5 to 0.05 in regards to annotation based metrics. Naturally, both more predictions are present and smaller kernel fragments appear. However,

## 4. Results

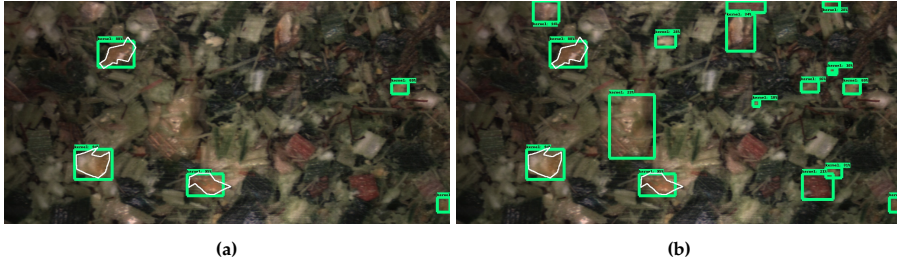


**Fig. B.7:** Bland-Altman plots for the baseline Faster R-CNN (a), Faster R-CNN 2a (b) and Faster R-CNN 2a with a confidence threshold of 0.05 (c). CSPS estimated with pixel distribution for baseline Faster R-CNN (d), Faster R-CNN 2a (e) and Faster R-CNN 2a with a confidence threshold of 0.05 (f).

we see that lowering the confidence threshold does not improve the precision but does increase the recall. The annotation process is both cumbersome and requires expert knowledge. Despite the experts annotating to the best of their abilities we have observed variation across annotators over numerous metrics including instances per image and average size of the instances. Therefore, we hypothesise that as smaller kernel fragments are only annotated in some cases the models struggle to optimise towards localising them, resulting in lower confidence for these predictions. However, as our images are captured in a largely controlled environment lowering the confidence does not increase the number of true false positives but only the annotated false positives. This result enhances the requirement of evaluating models in cluttered environments not only with metrics such as AP but also with other means independent of human annotation.

**Table B.7:** Precision/Recall at different confidence thresholds for the Faster R-CNN 2a model.

Conf. thresh.	AP@0.5	AR@0.5
0.5	56.40	56.30
0.25	42.79	69.31
0.05	22.43	81.43
0.005	8.83	88.21
0.0005	4.93	89.31



**Fig. B.8:** Example kernel fragment predictions from the 2a model with confidence threshold 0.5 (a) and 0.05 (b). Three fragments are annotated shown with the white outline significantly lowering the precision in (b) due to incorrectly annotated false positives.

**Table B.8:** AP results for overlength models trained on one of the four datasets. Each model is evaluated on the four test sets in the different column blocks.

Model	All			Small			Medium			Large		
	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75
<i>All</i>	<b>26.0</b>	<b>47.7</b>	<b>25.4</b>	28.0	52.9	26.2	<b>37.9</b>	<b>58.2</b>	<b>46.4</b>	<b>40.5</b>	<b>52.6</b>	<b>49.5</b>
<i>Small</i>	19.8	34.1	18.7	<b>32.2</b>	<b>56.2</b>	<b>32.3</b>	25.5	36.4	30.0	1.7	3.5	1.4
<i>Medium</i>	17.3	26.7	26.7	14.4	27.4	13.0	36.5	55.2	45.4	9.1	19.6	9.6
<i>Large</i>	7.6	12.6	7.2	2.5	6.5	1.7	10.1	23.5	8.0	27.9	41.9	34.5

## 4.2 Overlengths

In this section, we discuss the results from the Faster R-CNN variants trained for overlength recognition. This includes the effect of training data, filtering of predictions based upon the overlength definition, and anchor-tuning with k-means in the RPN.

## 4.3 Specialist models

First we explore the effect of the training data an overlength model is trained on, aiming to evaluate if specialist models can be created by training on a smaller dataset corresponding to the given verbal TLOC. In Table B.8 we show the AP results for the four test sets in each column block for Faster R-CNN models finetuned from COCO on the training sets *All*, *Small*, *Medium*, or *Large*. When testing on *All* the best performing model is trained on the larger *All* dataset containing all of the data from *Small*, *Medium* and *Large*. For specific verbal TLOCs we see the model trained only on the *Small* dataset is the best in terms of AP, when testing on corresponding sizes, improving by 4.2 pp in comparison to training on *All*. The result of specialist models on corresponding object sizes does not extend to the *Medium* and *Large* dataset. The *Medium*-trained models performs 1.4 pp worse for AP and we see for the *Large* test set the *Large*-trained model scores 12.6 pp lower than the *All* model.

The results are different when evaluating OVPS against the physical samples as shown in Table B.9. For both verbal TLOCs the Faster R-CNN trained

## 4. Results

in *Medium* has the highest correlation scores for PCC and  $r^2$ , however RMSE and CCC show better results for the models trained on the corresponding dataset. At verbal TLOC 4 mm there is a minor improvement compared to training on *All* or *Small*, however, at 12 mm there is significant improvement in both PCC and  $r^2$ .

**Table B.9:** OVPS correlation results for models trained on one of the four datasets against a verbal TLOC of 4 mm and 12 mm.

	Verbal TLOC 4 mm				Verbal TLOC 12 mm			
Model	PCC	$r^2$	RMSE	CCC	PCC	$r^2$	RMSE	CCC
<i>All</i>	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
<i>Small</i>	0.95	0.90	<b>14.92</b>	<b>0.58</b>	0.41	0.17	51.92	0.02
<i>Medium</i>	<b>0.97</b>	<b>0.93</b>	31.13	0.17	<b>0.77</b>	<b>0.60</b>	21.42	0.10
<i>Large</i>	0.86	0.74	35.09	0.07	0.53	0.28	<b>11.79</b>	<b>0.13</b>

In summary, for specialist models we see that when evaluating against annotations a model trained on all of the data performs best except for the *small* test set. Different results are seen against the physical OVPS where the *medium*-trained model is best, especially at verbal TLOC 12 mm.

### 4.4 Filtering

While we have used the overlength definition of  $1.5\times$  verbal TLOC during annotation the models trained on the *All* dataset do not have this information directly in the model and may not be able to make this distinction when predicting on image captured with a given verbal TLOC. Table B.10 shows the difference in AP for the *All* model when predictions for the given test set are filtered. We see minimal or no change in AP for both the *Small* and *Medium* test set but a significant improvement for the *Large* test set. In this case we see an increase of 5.4 pp for AP and 9.6 pp for AP@0.5 IoU. We hypothesise that this is due to the skew between the three datasets with *Large* having significantly less instances. Additionally, filtering in the *Small* test set is arguably not relevant as the *All* trained model is never shown overlength instances below this size.

Table B.11 shows the effect against the physical samples after filtering an *All-trained model*. We see slight improvements on the already highly correlated verbal TLOC 4 mm samples and a decrease at 12 mm for PCC and  $r^2$ . A surprising result as the AP results for the *Large* test set in Table B.10 saw significant improvements.

With filtering we can conclude from our data the results considerably improve for the large test set when evaluating against annotations, however, show little improvement at verbal TLOC 4 mm and decreases at 12 mm against OVPS correlation.

**Table B.10:** Precision results for the *All* model after filtering based on the overlength definition for the three sized test sets. On each test set we show the difference in the correlation against the model without filtering.

Test set	AP	AP@0.5	AP@0.75
<i>Small</i>	27.9	52.6	26.1
	-0.1	-0.3	-0.1
<i>Medium</i>	37.8	58.2	46.4
	-0.1	+0.0	+0.0
<i>Large</i>	45.9	62.2	55.5
	+5.4	+9.6	+6.0

**Table B.11:** OVPS correlation for the *All*-trained model without and with filtering applied in a post-processing step.

Model	Verbal TLOC 4 mm				Verbal TLOC 12 mm			
	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
<i>All</i>	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
<i>All Filter</i>	0.97	0.93	17.57	0.49	0.49	0.24	14.58	0.12

## 4.5 Transfer Learning

In Table B.12 we show the results for the baseline *All* model compared to models trained on *Small* with either transfer learning from COCO or from the *All*-trained model on the respective three sized test sets. In this case of finetuning from *All* we show the model with the respective verbal verbal TLOC dataset followed by  $_{ftall}$  or from COCO as  $_{coco}$ .

We see increases in AP on both the *Small* and *Medium* test sets when finetuning from the *All* model. The  $Small_{ftall}$  model increases by 2.1 pp to 30.1% but still performs worse than the *Small* model finetuned from COCO. It appears that the AP decreases at a lower IoU as the AP@0.5 decreases by 3.7 pp whereas AP@0.75 increases by 6.4 pp. A different trend appears for the *Medium*-tested models where the  $Medium_{ftall}$  sees a significant improvement in comparison to the two COCO finetuned models. Finetuning from *All* improves AP by 5.4 pp to 43.3% and similar increases are seen at the two shown IoU thresholds. Finally, the  $Large_{ftall}$  model does see a significant improvement when finetuning in comparison to the *Large* model finetuned from COCO, but still performs slightly worse than the *All* trained model on AP but with the exception of AP@0.5 increasing by 3.0 pp.

Table B.13 shows that adopting the respectively trained size models finetuned from an *All* model on for either verbal TLOC gives an improvement especially at 12 mm. The correlation is stronger compared to the *Medium* trained model finetuned from COCO, whilst PCC and r<sup>2</sup> improves by 0.23

## 4. Results

**Table B.12:** Precision results for different finetuning strategies. Each block evaluates on the test set that matches the given specialist model.

Model	AP	AP@0.5	AP@0.75
<i>All</i>	28.0	52.9	26.2
<i>Small</i>	32.2	<b>56.2</b>	32.3
<i>Small<sub>ftall</sub></i>	30.1	49.2	<b>32.6</b>
<i>All</i>	37.9	58.2	46.4
<i>Medium</i>	36.5	55.2	45.4
<i>Medium<sub>ftall</sub></i>	<b>43.3</b>	<b>63.0</b>	<b>50.4</b>
<i>All</i>	<b>40.5</b>	52.6	<b>49.5</b>
<i>Large</i>	27.9	52.6	<b>49.5</b>
<i>Large<sub>ftall</sub></i>	38.6	<b>55.6</b>	43.6

and 0.32 respectively. Additionally, RMSE decreases to 2.95 and CCC is relatively strong at 0.67.

**Table B.13:** OVPS correlation for different finetuning strategies. We include the previously well performing *Medium* specialist model and show results when training with respective specialists (Resp) for finetuning from COCO and *All*.

Model	Verbal TLOC 4 mm				Verbal TLOC 12 mm			
	PCC	r <sup>2</sup>	RMSE	CCC	PCC	r <sup>2</sup>	RMSE	CCC
<i>All</i>	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
<i>Medium<sub>coco</sub></i>	<b>0.97</b>	<b>0.93</b>	31.13	0.17	0.77	0.60	21.42	0.10
<i>Medium<sub>ftall</sub></i>	<b>0.97</b>	<b>0.93</b>	26.67	0.27	0.63	0.40	29.73	0.06
<i>Resp<sub>coco</sub></i>	0.95	0.90	<b>14.12</b>	<b>0.58</b>	0.53	0.28	11.79	0.13
<i>Resp<sub>ftall</sub></i>	0.96	<b>0.93</b>	19.63	0.43	<b>0.81</b>	<b>0.66</b>	<b>2.95</b>	<b>0.67</b>

The transfer learning results show different conclusions again based on the evaluation method. For AP we see that for the specialist models that small is best finetuned from COCO, medium from *All* and large shows better results at AP@0.5 from *All* but overall training with all annotations from COCO is best. From OVPS correlation we see that taking the respective specialist model finetuned from *All* provides the highest correlation.

### 4.6 Anchor tuning

Table B.14 provides a summary of the anchor tuning results for the three sized test sets. The AP results for each test set are grouped in a column with a model naming denoting either being trained on *All* or on the corresponding test set by the given specialist. Finally, for the *All* models we show the AP

results when filtering the predictions below the overlength definition for the given test set.

For the *Small* test set we see improvements for nearly all models across the AP metrics. Regardless of finetuning strategy, specialist models with anchor-tuning gives models that perform best, with models having two anchors improving AP by 5.1 pp when finetuning from COCO and by 5.7 pp when finetuning from *All*. There does appear to be a difference in how well the bounding-boxes fit the predictions objects dependent on the finetuning strategy. Models finetuned from *All* generally have a higher AP@0.5 where the best model increases by 5.6 pp, 2.7 pp more than the next best COCO-tuned model. However, for AP@0.75 COCO models perform better where the best result in *Specialist*<sub>12a</sub> increases by 12.9 pp compared to 6.1 pp. Table B.14 also shows a difference between the finetuning strategies for the *Medium* test set. COCO-tuned models trained on *All* perform better than the specialist *Medium* variants. However, when finetuning from *All* the specialist models outperform, with 6 anchors scoring highest across all three metrics, increasing AP by 8.4 pp, AP@0.5 by 5.7 pp and AP@0.75 by 10.5 pp. Again, for the *Large* test set we see difference between finetuning in Table B.14. *Large* trained anchor-tuned models that are finetuned from *All* perform much better than the counterparts finetuned from COCO. Regardless of finetuning strategy the models trained on *All* are best in terms of AP, with 2 anchors appearing to be the optimal for the *Large* sized test set. The model with 2 anchors and where predictions are filtered improve the AP by 18.3 pp to 58.8% with similar increased in pp to both IoU thresholds shown.

**Table B.14:** Anchor tuning results for variants of specialist models and differing finetuning strategies. The three columns show the three sized test sets where precision results either match an *All*-trained or corresponding sized dataset specialist-trained model. Each model is trained and evaluated with either 2a, 6a or 12a anchors, additionally the *All* models also have filtering applied. Finally, the models are grouped in rows by either finetuning on COCO or from an *All*-trained model.

Model	<i>Small</i>			<i>Medium</i>			<i>Large</i>		
	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75
<i>All</i>	28.0	52.9	26.2	37.9	58.2	46.4	40.5	52.6	49.5
<i>Finetune COCO</i>									
<i>All</i> <sub>2a</sub>	29.8	48.7	33.0	43.3	54.4	47.4	51.5	60.9	58.1
<i>All</i> <sub>2a</sub> <i>filter</i>	30.0	49.0	33.2	43.6	55.0	47.5	<b>58.8</b>	<b>71.3</b>	<b>66.0</b>
<i>All</i> <sub>6a</sub>	29.1	52.2	26.4	39.0	54.2	45.1	42.1	55.2	46.1
<i>All</i> <sub>6a</sub> <i>filter</i>	29.0	51.9	26.5	39.8	55.5	46.1	50.2	68.2	53.1
<i>All</i> <sub>12a</sub>	31.4	56.9	32.3	43.0	61.1	51.2	31.5	45.6	33.8
<i>All</i> <sub>12a</sub> <i>filter</i>	31.3	56.6	32.2	43.8	62.5	52.1	28.5	41.2	30.3
<i>Specialist</i> <sub>2a</sub>	33.1	<b>58.5</b>	32.1	34.7	54.3	38.5	18.8	31.8	20.7
<i>Specialist</i> <sub>6a</sub>	32.4	55.8	30.2	36.4	55.3	44.0	20.6	37.3	20.9
<i>Specialist</i> <sub>12a</sub>	30.8	52.7	31.8	37.1	55.1	44.8	19.9	34.0	14.7
<i>Finetune All</i>									
<i>All</i> <sub>2a</sub>	27.4	47.6	30.6	40.4	53.9	43.6	46.5	59.2	53.7
<i>All</i> <sub>2a</sub> <i>filter</i>	27.2	46.9	30.5	41.0	55.1	44.1	49.3	63.7	56.9
<i>All</i> <sub>6a</sub>	31.7	54.9	34.1	36.4	55.3	44.0	28.3	40.2	27.3
<i>All</i> <sub>6a</sub> <i>filter</i>	31.5	54.5	33.9	38.6	54.8	43.0	46.0	60.7	45.3
<i>All</i> <sub>12a</sub>	28.8	52.2	29.6	35.7	48.8	43.8	28.5	41.3	30.6
<i>All</i> <sub>12a</sub> <i>filter</i>	28.8	51.7	29.7	36.6	50.4	44.9	40.7	58.6	45.1
<i>Specialist</i> <sub>2a</sub>	<b>33.7</b>	55.8	37.4	42.7	60.2	52.1	24.5	41.0	30.6
<i>Specialist</i> <sub>6a</sub>	31.7	52.7	33.7	<b>46.3</b>	<b>63.9</b>	<b>56.9</b>	34.6	55.4	38.7
<i>Specialist</i> <sub>12a</sub>	33.5	54.7	<b>39.1</b>	38.8	55.2	49.2	39.5	57.7	47.6



#### 4. Results

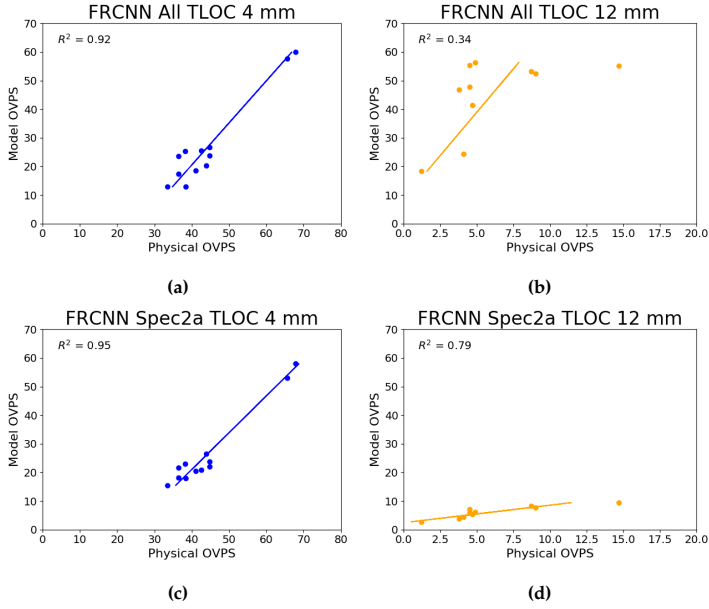
In Table B.15 we show the correlation results for the models and see again different trends compared to evaluating with annotations. The best performing model is when using the respective specialist finetuned from *All* for both verbal TLOCs. Again, only slight improvements can be made at 4 mm but at 12 mm a very strong correlation can be seen with both PCC and  $r^2$ . By having two anchors tuned for the overlength task as verbal TLOC 12 mm we improve the corresponding non-anchor tuned models in Table B.13 by 0.08 PCC and 0.13  $r^2$ . Additionally, the best performing models have an accompanying PCC p-value of 9.02e-8 and 0.00061 for the two verbal TLOCs. Similarly to previous correlation results the models with the highest PCC and  $r^2$  do not necessarily show the lowest RMSE or highest agreeance based on CCC.

**Table B.15:** OVPS correlation for the various anchor tuning models also shown in Table B.14.

	Verbal TLOC 4 mm				Verbal TLOC 12 mm			
Model	PCC	$r^2$	RMSE	CCC	PCC	$r^2$	RMSE	CCC
<i>All</i>	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
<i>Finetune COCO</i>								
<i>All2a</i>	0.97	0.94	22.62	0.30	0.67	0.45	27.23	0.05
<i>All2afilter</i>	0.97	0.94	22.65	0.30	0.41	0.17	26.05	0.12
<i>All6a</i>	0.96	0.91	19.51	0.36	0.66	0.44	29.76	0.04
<i>All6afilter</i>	0.96	0.91	19.63	0.36	0.50	0.25	7.57	0.25
<i>All12a</i>	0.96	0.92	18.83	0.42	0.63	0.40	33.58	0.03
<i>All12afilter</i>	0.96	0.92	18.86	0.42	0.54	0.29	5.16	0.37
<i>Specialist2a</i>	0.94	0.89	14.51	0.55	0.77	0.59	7.17	0.26
<i>Specialist6a</i>	0.95	0.89	<b>14.42</b>	<b>0.56</b>	0.63	0.40	7.51	0.24
<i>Specialist12a</i>	0.94	0.88	16.19	0.50	0.81	0.65	7.24	0.26
<i>Finetune All</i>								
<i>All2a</i>	0.93	0.87	24.14	0.24	0.62	0.38	17.6	0.15
<i>All2afilter</i>	0.93	0.87	24.18	0.24	0.51	0.26	6.02	0.31
<i>All6a</i>	0.96	0.91	22.39	0.33	0.74	0.54	22.14	0.12
<i>All6afilter</i>	0.96	0.91	22.43	0.33	0.53	0.29	<b>4.67</b>	0.41
<i>All12a</i>	0.91	0.83	25.31	0.20	0.77	0.59	13.8	0.09
<i>All12afilter</i>	0.91	0.83	25.40	0.20	0.68	0.46	5.69	0.42
<i>Specialist2a</i>	<b>0.97</b>	<b>0.95</b>	18.11	0.45	<b>0.89</b>	<b>0.79</b>	7.24	0.21
<i>Specialist6a</i>	<b>0.97</b>	0.93	20.17	0.40	0.83	0.69	5.41	<b>0.45</b>
<i>Specialist12a</i>	0.96	0.92	19.06	0.41	0.73	0.53	6.69	0.27

From the large number of models trained for overlengths we see the best results for both AP and correlation with anchor tuning. When evaluating AP we see that specialist models with either 2 or 12 anchors perform best on the small test set, whereas for medium the specialist 6 anchors finetuned from *All* scores highest and for large the *All* filtered with 2 anchors is best. However, we see consistent results for OVPS correlation for a *All*-finetuned respective specialist model with 2 anchors. In addition to the highest correlated across all anchor tuning models it is also the strategy with highest OVPS correlation across all results in this section.

Finally, corresponding Bland-Altman plots for models in Figure B.9 are

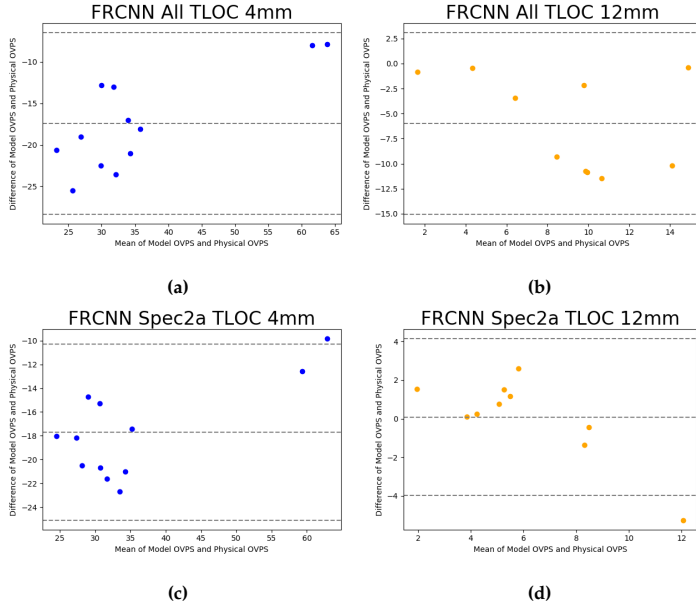


**Fig. B.9:** OVPS correlation for different verbal TLOCs. The *All*-trained model at 4 mm (a) and 12 mm (b). Additionally, the best performing model in terms in PCC and  $r^2$  Specialist2a finetuned from *All* at verbal TLOC 4 mm (c) and 12 mm (d).

shown in Figure B.10. At verbal TLOC 4 mm the analysis between the two models are largely similar, whereas at verbal TLOC 12 mm agreeance is larger with points clustering closer together and an overall lower difference between model estimates and physical measurement.

Lastly, we show example predictions from the three best performing models for the respective sized test sets in Figures B.11-B.13. We see high amounts of precision, as shown in previous AP results, for each of the four classes with predicted bounding-boxes being sized appropriately. In addition, we see that the models have learnt to distinguish between similar classes such as accepted and non-accepted leaves where only the fibre structure is the clearest difference between the two.

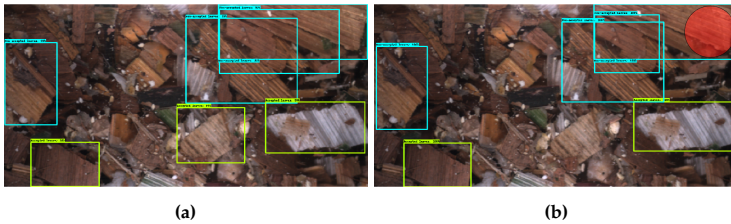
## 4. Results



**Fig. B.10:** Bland-Altman plots for different verbal TLOCs. The *All*-trained model at 4 mm (a) and 12 mm (b). Additionally, the best performing model in terms in PCC and  $r^2$  Specialist2a finetuned from *All* at verbal TLOC 4 mm (c) and 12 mm (d).



**Fig. B.11:** Example predictions from the Specialist2a finetuned from *All* on images from the *Small* test set. Predictions (a) and ground truth (b).



**Fig. B.12:** Example predictions from the Specialist2a finetuned from *All* on images from the *Medium* test set. Predictions (a) and ground truth (b).



**Fig. B.13:** Example predictions from the Specialist2a finetuned from *All* on images from the *Large* test set. Predictions (a) and ground truth (b).

## 5 Discussion

### 5.1 Annotation Data

The increased correlation for CSPA found for Faster R-CNN after lowering the confidence threshold for predictions to between 0.05 to 0.005 is in stark contrast to the typical threshold of at least 0.5, for example, in the TensorFlow object detection API [38], when visualising the standard threshold is 0.5. However, with a subjective inspection of predictions at such a low threshold there does not appear to be a large number of actual false positives but rather instances that were not annotated. This is likely due to the challenging annotation process and differing interpretations of kernel fragments between annotators. Future care could be taken to re-evaluate the annotations in order to improve the data integrity, this process could be done manually or by more automated approaches such as semi-supervised or active learning. After this process it would be fair to hypothesise that the computer vision metrics would improve, however, of more interest is if the already strong correlation would increase. While the annotation process is challenging and this results in the requirement of lowering the confidence threshold the anchor-tuned Faster R-CNN still appear to capture the overall particle size distribution and thereby CSPA well. Therefore, it can be argued that a lower confidence threshold is acceptable at a less precise annotated dataset.

### 5.2 Algorithm Verification

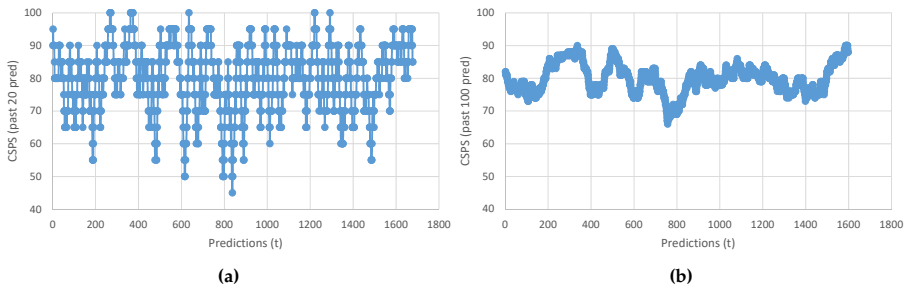
In this work we evaluated the anchor-tuning method on a Faster R-CNN network. However, it would be of interest to see if this translates to other two-stage networks, either with a different meta-architecture such as Mask R-CNN [41] or with a different backbone feature extractor such as ResNets [42] or MobileNets [43]. Our initial analysis with a Mask R-CNN and Inceptionv2 show similar trends to what we have shown in this work, however, our anchor-tuned Faster R-CNN still outperforms these networks. Espe-

cially of interest would be an overall architecture with lower requirements on complexity opening an opportunity for an embedded prototype for farmers. While embedded hardware exists with large amounts of processing power to run a Faster R-CNN with Inceptionv2, investigating if CSPS and OVPS correlate with a lower complexity model would be beneficial. A number of additional options exist in TensorFlow object detection API as explored in [9] and additional are present in the API with TensorFlow version 2. Alternatively a custom architecture specifically for the task optimised with TensorFlow lite could also be an option.

### 5.3 Automating Corn Silage Harvesting

In this work we have shown that with tuned Faster R-CNNs there is a strong correlation between model predictions and the quality of physically measured corn silage. However, there are still a number of open questions before these models can be deployed in the field.

Firstly, we compute the CSPS/OVPS over all predictions in a image set, however, in the field the models would be within a system giving feedback to how to alter the PG and verbal TLOC. Therefore, work is required to determine how many images are required before the models are giving a consistent signal that can be used. An example of this problem is visualised in Figure B.14 where we show the CSPS computed for one of our image sets for either the past 20 predictions (a) or 100 predictions (b). The best performing model in terms of physical correlation found a CSPS score of 80.38 over the 1697 predictions. But as can be seen in Figure B.14 the CSPS calculation is quite unstable with a rolling average of 20 predictions varying between 45 to 100 CSPS. When increasing the number of predictions to the past 100 the signal is more stable but additional work is required to determine the optimal range.



**Fig. B.14:** Rolling average for CSPS for an image set for the past 20 predictions (a) and past 100 predictions (b). The best performing model found in this work computed a CSPS of 80.38 over the entire image set.

Further work is also required in creating a prototype system to suggest

how much the PG and verbal TLOC should be altered given the model predictions. As covered in Section F.1 suboptimal machine settings can effect resulting milk yield from dairy cows, promote unwanted bacteria during packing or result in wasted fuel consumption. Future steps will be to address the model signal can be used in a system to optimise towards farmer requirements on for their corn silage.

## 6 Conclusion

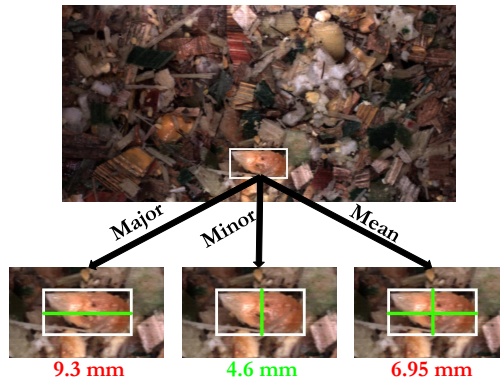
We present the first work to automate the measurement of chopped stover from corn silage harvested with a forage harvester. Our work predicts the instances of overlengths which is dependent on a farmer's desired stover length based upon the verbal TLOC. Additionally, we show improvements to previous methods of measuring kernel fragmentation in images of non-separated corn silage. Experimental evaluation on a number of strategies for model development led to significant improvements compared to a naive object detection training methodology for both tasks. We evaluated the effect of training specialist models towards a given verbal TLOC on a subset of the data. We experimented with two transfer learning strategies showing that in general better performance was found when a model was finetuned from all verbal TLOCs rather than a significantly different domain. Finally, we showed that anchor tuning significantly improved performance for both tasks. Our approaches led to model improvements of up to 45.2% for AP. However, evaluation against physical measurements indicated potential flaws when only validating models with manual annotations, thereby showing that model development and testing should be done in a complementary manner between the two. Despite the challenging annotations, general trends for improved models were seen for both evaluation purposes and the annotations led to strong correlations in the models.

## 7 Appendix

In this appendix we include additional investigations to those covered earlier and published in our journal article. We investigate the effect of classifying kernel particles based upon different axes. Additionally, for both tasks, different feature extractors are evaluated for Faster R-CNNs. The difference in determining CSPS and OVPS between Faster R-CNN and Mask R-CNN is shown. Finally, the effect of lowering the image resolution is covered for both tasks.

### 7.1 CSPS Classification

In this work Faster R-CNNs [12] with Inceptionv2 [44] backbones were trained to predict kernel fragments across an image set with varying machine settings. For each prediction, the major axis of the bounding-box was compared against the CSPS threshold of 4.75 mm, where if the axis length is below the threshold the prediction is deemed as sufficiently processed. Our approach aims to mimic a sieving system, predicting if a fragment would fall through a sieve or not. However, sieving often shakes the sample in three dimensions and a particle may rotate such that it passes a sieve based on the minor axis. The difference in CSPS classification is visualised in Figure B.15 for an example fragment prediction localised with a bounding-box. Here, the same fragment can be classified based upon a major axis length of 9.3 mm, compared to a minor axis of 4.6 mm, or as an alternative, the mean of the two of 6.95 mm. In this example, if the major or mean axis is used the fragment is classified as insufficiently fragmented, however, based on the minor axis it is deemed sufficient.



**Fig. B.15:** Different axis lengths can be used to classify a predicted kernel fragment from a bounding-box detector. Using the major or mean axis classifies fragments as insufficiently processed with the CSPS threshold, whereas the minor classifies it as sufficiently processed.

To evaluate the optimal axis for classification, we again train Faster R-CNN [12] with Inceptionv2 [44] models with a baseline, and additionally tuned with 2 anchors that are evaluated at a confidence threshold of 0.05 (2a). In Table B.16 the correlation results across the harvest weeks are shown. As in Section 4.1, we evaluate the estimated CSPS based on counting the percentage of particles (counts) and based on the percentage of bounding-box pixels below the CSPS threshold (size). The table shows that the results are not improved to that covered in the original paper, where the 2a models with CSPS estimated based on bounding-box sizes and classified with the major axis performs best. In both CSPS estimation approaches the correlation decreases between major and minor axis classification, especially when estimating CSPS on instance counts. Taking the mean of the two performs slightly worse than using the major axis.

**Table B.16:** Correlation results for models where a fragment is classified with either the major, minor or mean bounding-box axis. Baseline is a Faster R-CNN Inceptionv2 and 2a is a Faster R-CNN Inceptionv2 with 2 anchors tuned and a confidence threshold of 0.05.

		CW40		CW43		CW40+CW43	
Model	Axis	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>
<i>Counts</i>							
Baseline	Major	0.80	0.64	0.72	0.52	0.63	0.40
2a	Major	0.94	0.87	0.78	0.61	0.80	<b>0.65</b>
2a	Minor	0.83	0.70	0.74	0.55	0.74	0.54
2a	Mean	0.92	0.85	0.71	0.51	0.77	0.59
<i>Size</i>							
Baseline	Major	0.87	0.75	0.74	0.55	0.73	0.53
2a	Major	<b>0.95</b>	<b>0.91</b>	<b>0.79</b>	<b>0.63</b>	<b>0.81</b>	<b>0.65</b>
2a	Minor	0.88	0.77	0.76	0.57	0.74	0.55
2a	Mean	0.93	0.87	0.72	0.52	0.76	0.58

## 7.2 Feature extractor

In our original work we only evaluated object detectors with an Inceptionv2 [44] backbone, however, other options exist in the TensorFlow Object Detection API [38]. Therefore, we trained Faster R-CNN [12] with a ResNet50 [45] to determine the difference between two feature extractors. The two models have similar complexity in terms of parameters but have different architectures. A ResNet is built around skip connections passing a residual function and an Inception module is wider consisting of a number of different sized convolutional filters. The ResNet50-based models are also trained with and without 2 anchors tuned for the task. Following the baseline parameters in



## 7. Appendix

the TensorFlow API [38], the models were optimised using stochastic gradient descent with a learning rate of 0.003, momentum of 0.9 and a batch size of 1. A total of 25000 training iterations was conducted and the model with the lowest validation loss was taken for testing.

In Table B.17 the AP results for the trained models compared to those already presented with Inceptionv2 are shown. With a ResNet50 backbone the AP results are not improved, however, there is a similar trend when anchor tuning is added.

**Table B.17:** AP results comparing Faster R-CNN (FRCNN) models with an Inceptionv2 (Iv2) and ResNet50 (R50) backbone. The 2a models are with two anchors tuned.

Model	AP	AP@0.5	AP@0.75
FRCNN Iv2	25.6	51.9	22.3
FRCNN Iv2 2a	<b>28.5</b>	<b>56.6</b>	<b>25.7</b>
FRCNN R50	24.5	51.1	20.4
FRCNN R50 2a	25.3	51.5	21.7

In Table B.18 correlation analysis for models are evaluated at a confidence threshold of 0.5 and 0.05. Following the results in the previous section the fragments are only classified based on the major axis. The results can be seen in Table B.18 for both CSPS on instance counts and bounding-box size. In the table we see that again the model with the highest correlation is still that determined in Section 4.1. As seen for AP, similar trends can also be seen for the ResNet50-based models, in that correlation increases when lowering the confidence threshold from 0.5 to 0.05 and initialising the RPN with 2 anchors.

**Table B.18:** Training Faster R-CNN (FRCNN) with an Inceptionv2 (Iv2) or ResNet50 (R50) shows differences in correlation. Additionally, the confidence threshold for predictions are shown for 0.5 and 0.05.

Model	CW40		CW43		CW40+CW43	
	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>
<i>Counts</i>						
FRCNN Iv2 2a c0.05	0.94	0.87	0.78	0.61	0.80	0.65
FRCNN R50 c0.5	0.78	0.61	0.68	0.46	0.65	0.43
FRCNN R50 c0.05	0.87	0.77	0.66	0.44	0.72	0.51
FRCNN R50 2a c0.5	0.82	0.67	0.61	0.37	0.49	0.24
FRCNN R50 2a c0.05	0.89	0.79	0.67	0.45	0.54	0.29
<i>Size</i>						
FRCNN Iv2 2a c0.05	<b>0.95</b>	<b>0.91</b>	<b>0.79</b>	<b>0.63</b>	<b>0.81</b>	<b>0.65</b>
FRCNN R50 c0.5	0.74	0.54	0.73	0.53	0.68	0.46
FRCNN R50 c0.05	0.88	0.77	0.66	0.44	0.73	0.53
FRCNN R50 2a c0.5	0.81	0.66	0.65	0.42	0.67	0.32
FRCNN R50 2a c0.05	0.90	0.81	0.63	0.39	0.56	0.32

For corresponding stover overlengths models we again see in Table B.19 that better performing models are in general found with an Inceptionv2 backbone. Similar to Inceptionv2, the ResNet backbone has a decrease in AP when training a specialist model with anchor tuning.

**Table B.19:** AP results comparing Faster R-CNN with different feature extractors.

Model	<i>Small</i>			<i>Large</i>		
	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75
<i>Train All</i>						
FRCNN Iv2	28.0	52.9	26.2	<b>40.5</b>	<b>52.6</b>	<b>49.5</b>
FRCNN R50	26.2	48.6	26.4	32.4	51.0	41.0
<i>Train Specialist</i>						
FRCNN Iv2 2a <sub>coco</sub>	33.1	<b>58.5</b>	32.1	18.8	31.8	20.7
FRCNN Iv2 2a <sub>all</sub>	<b>33.7</b>	55.8	<b>37.4</b>	24.5	41.0	30.6
FRCNN R50 2a <sub>coco</sub>	28.1	47.8	30.4	16.3	32.3	10.9
FRCNN R50 2a <sub>all</sub>	29.6	50.7	31.8	23.6	39.6	23.9

Table B.20 shows the correlation results for different feature extractors. With ResNet50 there is not as large an increase in the correlation when tuning the models compared to a baseline. Overall, the best performing models is still a specialist model with 2 anchors and finetuned from *All*.

**Table B.20:** Faster R-CNN (FRCNN) correlation with different feature extractors, namely Inceptionv2 (Iv2) and ResNet50 (R50), for overlength models at two TLOCs.

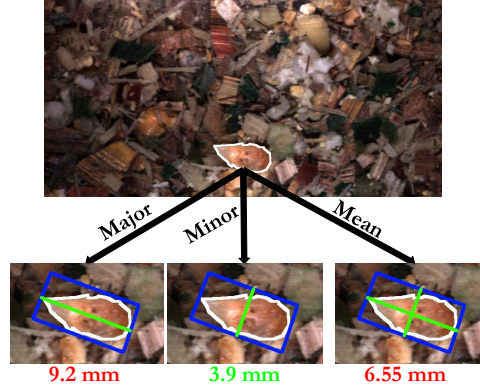
Model	TLOC 4 mm		TLOC 12 mm	
	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>
<i>Train All</i>				
FRCNN Iv2	0.96	0.92	0.58	0.34
FRCNN R50	0.94	0.89	0.75	0.57
<i>Train Specialist</i>				
FRCNN Iv2 2a <sub>coco</sub>	0.94	0.89	0.77	0.59
FRCNN Iv2 2a <sub>all</sub>	<b>0.97</b>	<b>0.95</b>	<b>0.89</b>	<b>0.79</b>
FRCNN R50 2a <sub>coco</sub>	0.93	0.87	0.77	0.59
FRCNN R50 2a <sub>all</sub>	0.96	0.92	0.72	0.52

### 7.3 Box vs Mask Localisation

Estimating the CSPS using predictions found with bounding-boxes can allow for less precise measurements as a box can overestimate the size in comparison to a segmentation mask. Figure B.15 shows this for a kernel prediction

## 7. Appendix

where the instance is slightly rotated. However, by predicting masks a more precise estimate of the size can be obtained as shown in Figure B.16. For the same fragment different axis lengths can be estimated as a rotated bounding-box can fit around the predicted mask. In this case each of the lengths are shorter than the bounding-box counterpart in Figure B.15.



**Fig. B.16:** Different axis lengths can be used to classify a predicted kernel fragment from a segmentation mask. Using the major or mean axis classifies fragments as insufficiently processed with the CSPS threshold, whereas the minor classifies it as sufficiently processed.

Therefore, Mask R-CNNs [41] with an Inceptionv2 [44] were trained, with and without 2 anchors tuned and evaluated at different confidence thresholds. Again, models were trained for 25000 iterations and the model with the lowest loss was used for testing. Training parameters were used from those provided in the API [38], where model optimisation was done with stochastic gradient descent with a learning rate of 0.002, momentum of 0.9 and batch size of 1.

In Table B.21 the difference in AP for kernel fragmentation between Faster R-CNN and Mask R-CNN is shown. AP is lower when predicting masks compared to bounding-boxes. Additionally, the Mask R-CNN model does not improve when adding anchor tuning.

**Table B.21:** AP results comparing Faster R-CNN (FRCNN) and Mask R-CNN (MRCNN) models with an Inceptionv2 (Iv2) backbone. The 2a models are with two anchors tuned.

Model	AP	AP@0.5	AP@0.75
FRCNN Iv2	25.6	51.9	22.3
FRCNN Iv2 2a	<b>28.5</b>	<b>56.6</b>	<b>25.7</b>
MRCNN Iv2	24.5	53.7	17.3
MRCNN Iv2 2a	23.3	51.5	16.1

Table B.22 shows the correlation results and in this case improvements can be seen when lowering the confidence threshold and adding anchor tuning compared to a baseline. As for the Faster R-CNN models in Section 7.1, the minor axis is the worst performing in terms of correlation. Additionally, despite the hypothesis that a segmentation mask would provide better localisation this is not seen in the correlation results. Possibly this can be explained by suboptimal training parameters or the Mask R-CNN is more challenging to train in comparison to the Faster R-CNN models.

**Table B.22:** Correlation results for Mask R-CNN (MRCNN) with Inceptionv2 (Iv2) when classifying fragments at different axis lengths.

		CW40		CW43		CW40+CW43	
Model	Axis	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>
Counts							
FRCNN Iv2 2a c0.05	Major	0.94	0.87	0.78	0.61	0.80	<b>0.65</b>
MRCNN Iv2 c0.5	Major	0.61	0.37	0.55	0.30	0.49	0.24
MRCNN Iv2 c0.05	Major	0.82	0.68	0.65	0.42	0.60	0.36
MRCNN Iv2 2a c0.5	Major	0.74	0.55	0.65	0.42	0.59	0.35
MRCNN Iv2 2a c0.05	Major	0.87	0.76	0.74	0.54	0.67	0.45
MRCNN Iv2 2a c0.5	Minor	0.77	0.59	0.74	0.54	0.66	0.43
MRCNN Iv2 2a c0.05	Minor	0.78	0.61	0.80	0.63	0.69	0.48
MRCNN Iv2 2a c0.5	Mean	0.82	0.68	0.74	0.53	0.70	0.49
MRCNN Iv2 2a c0.05	Mean	0.86	0.74	0.79	0.63	0.73	0.54
Size							
FRCNN Iv2 2a c0.05	Major	<b>0.95</b>	<b>0.91</b>	0.79	0.63	<b>0.81</b>	<b>0.65</b>
MRCNN Iv2 c0.5	Major	0.74	0.54	0.60	0.36	0.58	0.34
MRCNN Iv2 c0.05	Major	0.67	0.45	0.63	0.39	0.51	0.26
MRCNN Iv2 2a c0.5	Major	0.70	0.48	0.69	0.48	0.61	0.38
MRCNN Iv2 2a c0.05	Major	0.87	0.75	0.77	0.58	0.72	0.52
MRCNN Iv2 2a c0.5	Minor	0.78	0.61	0.78	0.62	0.68	0.47
MRCNN Iv2 2a c0.05	Minor	0.81	0.66	0.78	0.61	0.70	0.49
MRCNN Iv2 2a c0.5	Mean	0.82	0.67	0.77	0.59	0.73	0.53
MRCNN Iv2 2a c0.05	Mean	0.86	0.73	<b>0.81</b>	<b>0.66</b>	0.75	0.56

Mask R-CNN variants were trained in the same manner for OVPS and the AP results can be seen in Table B.23. As shown for kernel fragmentation, higher AP is seen with a Faster R-CNN. However, anchor tuning does provide some improvement in comparison to a baseline Mask R-CNN.

## 7. Appendix

**Table B.23:** AP results comparing Faster R-CNN (FRCNN) and Mask R-CNN (MRCNN).

Model	<i>Small</i>			<i>Large</i>		
	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75
<i>Train All</i>						
FRCNN Iv2	28.0	52.9	26.2	<b>40.5</b>	<b>52.6</b>	<b>49.5</b>
MRCNN Iv2	26.7	51.1	28.6	14.2	26.7	12.3
<i>Train Specialist</i>						
FRCNN Iv2 2a <sub>coco</sub>	33.1	<b>58.5</b>	32.1	18.8	31.8	20.7
FRCNN Iv2 2a <sub>all</sub>	<b>33.7</b>	55.8	<b>37.4</b>	24.5	41.0	30.6
MRCNN Iv2 2a <sub>coco</sub>	28.9	54.7	28.1	14.9	30.3	14.3
MRCNN Iv2 2a <sub>all</sub>	31.5	58.8	32.3	16.4	35.9	12.5

The OVPS correlation results for Mask R-CNN are shown in Table B.24. There is a decrease in PCC at TLOC 4 mm and a very slight increase in  $r^2$ . Whereas, for TLOC 12 mm there is an improvement in both correlation scores when using a Mask R-CNN and tuning 2 anchors. However, different from previous results, for TLOC 12 mm Mask R-CNN finetuning from COCO shows significantly higher correlation than from *All*.

**Table B.24:** Difference between Faster R-CNN (FRCNN) and Mask R-CNN (MRCNN) with Inceptionv2 (Iv2) for OVPS.

Model	TLOC 4 mm		TLOC 12 mm	
	PCC	$r^2$	PCC	$r^2$
<i>Train All</i>				
FRCNN Iv2	0.96	0.92	0.58	0.34
MRCNN Iv2	0.93	0.87	0.81	0.66
<i>Train Specialist</i>				
FRCNN Iv2 <sub>coco</sub>	0.93	0.87	0.81	0.66
FRCNN Iv2 2a <sub>all</sub>	<b>0.97</b>	<b>0.95</b>	0.89	0.79
MRCNN Iv2 2a <sub>coco</sub>	0.94	0.89	0.92	<b>0.85</b>
MRCNN Iv2 2a <sub>all</sub>	0.91	<b>0.96</b>	0.70	0.43

### 7.4 Image Resolution

In previous work [9], the image resolution was evaluated for a number of models. It was seen for the Faster R-CNN Inceptionv2 there were slight differences in AP and AR between an image resolution of 600x1200 and 400x730. We now extend this to include anchor tuning and perform a correlation analysis where models are trained and evaluated at three different image resolutions.

## Chapter B.

Table B.25 again shows the slight decrease from a resolution of 600x1200 to 400x730 for kernel fragmentation, however, a significant drop in AP is seen when at the smallest resolution. Additionally, for all image resolution combinations we see improvements in AP when applying anchor tuning.

**Table B.25:** AP results for kernel fragmentation at different image resolutions.

Model	AP	AP@0.5	AP@0.75	
FRCNN	600x1200	25.6	51.9	22.3
FRCNN 2a	600x1200	<b>28.5</b>	<b>56.6</b>	<b>25.7</b>
FRCNN	400x730	24.5	52.5	18.9
FRCNN 2a	400x730	26.9	55.1	22.8
FRCNN	200x365	15.7	39.4	9.0
FRCNN 2a	200x365	15.7	41.1	8.3

In Table B.26 the correlation results are shown and a significant decrease in PCC and  $r^2$  can be seen as the resolution is lowered.

**Table B.26:** CSPS correlation results for Faster R-CNN (FRNN) models trained and evaluated at three different image resolutions.

		CW40		CW43		CW40+CW43	
Model	Image Resolution	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>	PCC	r <sup>2</sup>
Counts							
FRCNN c0.5	600x1200	0.68	0.46	0.64	0.41	0.53	0.28
FRCNN c0.05	600x1200	0.80	0.64	0.72	0.52	0.63	0.40
FRCNN 2a c0.5	600x1200	0.84	0.70	0.63	0.40	0.64	0.40
FRCNN 2a c0.05	600x1200	0.94	0.87	0.78	0.61	0.80	<b>0.65</b>
Size							
FRCNN c0.5	600x1200	0.73	0.53	0.66	0.44	0.64	0.41
FRCNN c0.05	600x1200	0.87	0.75	0.74	0.55	0.73	0.53
FRCNN 2a c0.5	600x1200	0.79	0.62	0.70	0.48	0.66	0.43
FRCNN 2a c0.05	600x1200	<b>0.95</b>	<b>0.91</b>	<b>0.79</b>	<b>0.63</b>	<b>0.81</b>	<b>0.65</b>
Counts							
FRCNN c0.5	400x730	0.72	0.52	0.62	0.39	0.53	0.28
FRCNN c0.05	400x730	0.86	0.74	0.63	0.40	0.56	0.31
FRCNN 2a c0.5	400x730	0.50	0.25	0.55	0.30	0.46	0.21
FRCNN 2a c0.05	400x730	0.88	0.78	0.57	0.33	0.63	0.39
Size							
FRCNN c0.5	400x730	0.67	0.45	0.70	0.50	0.59	0.35
FRCNN c0.05	400x730	0.85	0.73	0.69	0.47	0.65	0.42
FRCNN 2a c0.5	400x730	0.54	0.29	0.57	0.33	0.49	0.24
FRCNN 2a c0.05	400x730	0.87	0.76	0.64	0.41	0.67	0.45
Counts							
FRCNN c0.5	200x365	0.78	0.60	0.67	0.45	0.55	0.31
FRCNN c0.05	200x365	0.76	0.58	0.62	0.39	0.60	0.36
FRCNN 2a c0.5	200x365	0.65	0.43	0.57	0.33	0.56	0.31
FRCNN 2a c0.05	200x365	0.80	0.63	0.61	0.37	0.59	0.35
Size							
FRCNN c0.5	200x365	0.68	0.47	0.60	0.36	0.47	0.22
FRCNN c0.05	200x365	0.76	0.58	0.62	0.39	0.60	0.36
FRCNN 2a c0.5	200x365	0.89	0.80	0.60	0.35	0.65	0.42
FRCNN 2a c0.05	200x365	0.78	0.60	0.65	0.43	0.66	0.43

We also show the timings for the detectors in Table B.27 on an NVIDIA Titan XP. It can be seen that for the average time over 100 images decreases

## 7. Appendix

as the image resolution becomes smaller. Additionally, using 2 anchors compared to 12 in the baseline also decreases the timings slightly.

**Table B.27:** Average time per image for models at the three different image resolutions.

Model	Image Resolution	Avg time (ms)
FRCNN Iv2	600x1200	0.052
FRCNN Iv2 2a	600x1200	0.046
FRCNN Iv2	400x730	0.044
FRCNN Iv2 2a	400x730	0.038
FRCNN Iv2	200x365	0.042
FRCNN Iv2 2a	200x365	0.039

In Table B.28 the AP results for stover overlength Faster R-CNNs can be seen for the three image resolutions. Interestingly, the trend that the *Large* test set is best when training on the *All* dataset is consistent for each image resolution. Furthermore, we seen an increase in AP when decreasing the resolution to 400x730.

**Table B.28:** AP results for overlength models at different image resolutions.

Model	<i>Small</i>			<i>Large</i>		
	AP	AP@0.5	AP@0.75	AP	AP@0.5	AP@0.75
<i>Train All</i>						
600x1200	28.0	52.9	26.2	<b>40.5</b>	<b>52.6</b>	<b>49.5</b>
400x730	25.4	44.9	24.5	51.4	66.7	61.7
200x365	19.1	38.9	17.2	41.0	53.3	46.9
<i>Train Specialist</i>						
600x1200	33.7	55.8	37.4	24.5	41.0	30.6
400x730	28.5	51.2	28.4	25.8	41.3	26.0
200x365	21.6	45.4	16.2	21.8	40.9	18.4

However, for OVPS correlation there is a decrease in the scores as the resolution is lowered as seen in Table B.29. There is a stark difference compared to the previous table, where we see for correlation *Specialist*-trained models significantly outperform the *All*-trained. As for earlier results, we hypothesize that this is due to the relatively small number of annotations for the *Large* test set.

**Table B.29:** OVPS correlation for different image resolutions. Models are all trained with tuning two anchors.

Image Resolution	TLOC 4 mm		TLOC 12 mm	
	PCC	$r^2$	PCC	$r^2$
<i>Train All</i>				
600x1200	0.96	0.92	0.58	0.34
400x730	0.94	0.89	0.36	0.13
200x365	0.92	0.85	0.41	0.17
<i>Train Specialist</i>				
600x1200	<b>0.97</b>	<b>0.95</b>	<b>0.89</b>	<b>0.79</b>
400x730	0.96	0.92	0.72	0.51
200x365	0.95	0.91	0.57	0.32

Finally, similar timings to kernels are seen in Table B.30 as the image resolution decreases for stover overlenghts. However, the timings are slightly higher possibly due to the larger number of classes present.

**Table B.30:** Timings at different image resolutions for stover overlenghts with Faster R-CNN (FRCNN).

Model	Image Resolution	Avg time (ms)
FRCNN Iv2	600x1200	0.053
FRCNN Iv2 2a	600x1200	0.048
FRCNN Iv2	400x730	0.052
FRCNN Iv2 2a	400x730	0.044
FRCNN Iv2	200x365	0.046
FRCNN Iv2 2a	200x365	0.044

## 7.5 Conclusion

In this appendix different design choices for two-stage networks are covered for recognition of kernel fragments and stover overlenghts. It is shown that for CSPS classifying kernel fragments based on the major axis outperform minor and mean for both Faster R-CNN and Mask R-CNN. In addition, bounding-boxes proved to perform better with Faster R-CNN instead of the more precise masks from Mask R-CNN for CSPS. However, Mask R-CNN for OVPS estimation did improve the correlation, especially as TLOC 12 mm.

It was also shown that the choice of feature extractor can be important. Faster R-CNNs with both Inceptionv2 and ResNet50 are evaluated, where the former provides better results by a number of percentage points in most cases.



## 7. Appendix

Finally, the image resolution is shown to have a significant effect on the correlation to the physical samples. In [9], the difference between 600x1200 and 400x730 image resolution was minimal with Inceptionv2 in terms of AP, however, for the estimated quality metrics the PCC and  $r^2$  dropped significantly.

These results show that for WPCS, tuning anchors in two-stage networks performs well with a number of design choices, however, care should be taken when determining the final architecture.

## References

- [1] J. Heinrichs and M. J. Coleen, "Penn state particle separator," May 2016. [Online]. Available: <https://extension.psu.edu/penn-state-particle-separator>
- [2] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [3] ASABE, "Method of determining and expressing particle size of chopped forage materials by screening," *ANSI/ASAE*, vol. S424.1, p. 663–665.
- [4] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," *In: Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [5] D. Mertens, "Creating a system for meeting the fiber requirements of dairy cows," *Journal of Dairy Science*, vol. 80, no. 7, pp. 1463–1481, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022030297760752>
- [6] J. L. Drewry, B. D. Luck, R. M. Willett, E. M. Rocha, and J. D. Harmon, "Predicting kernel processing score of harvested and processed corn silage via image processing techniques," *Computers and Electronics in Agriculture*, vol. 160, pp. 144 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918311955>
- [7] B. D. Luck, J. L. Drewry, R. D. Shaver, R. M. Willett, and L. F. Ferraretto, "Predicting in situ dry matter disappearance of chopped and processed corn kernels using image-analysis techniques," *Applied Animal Science*, vol. 36, no. 4, pp. 480–488, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590286520300847>
- [8] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [9] C. B. Rasmussen and T. B. Moeslund, "Evaluation of model selection for kernel fragment recognition in corn silage," <https://arxiv.org/abs/2004.00292>, 2020.
- [10] P. Savoie, M.-A. Audy-Dubé, G. Pilon, and R. Morissette, "Chopped forage particle size analysis in one, two and three dimensions," 01 2013.

## References

- [11] M. Audy, P. Savoie, F. Thibodeau, and R. Morissette, "Size and shape of forage particles by image analysis and normalized multiscale bending energy method," *American Society of Agricultural and Biological Engineers Annual International Meeting 2014, ASABE 2014*, vol. 2, pp. 820–830, 01 2014.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 06 2016.
- [14] H. Kaur and B. Singh, "Classification and grading rice using multi-class svm," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, pp. 1–5, 2013.
- [15] A. Aggarwal and R. Mohan, "Aspect ratio analysis using image processing for rice grain quality," *International Journal of Food Engineering*, vol. 6, 01 2010.
- [16] G. Dalen, "Determination of the size distribution and percentage of broken kernels of rice using flatbed scanning and image analysis," *Food Research International*, vol. 37, pp. 51–58, 06 2004.
- [17] P. Dubosclard, S. Larnier, H. Konik, A. Herbulot, and M. Devy, "Automatic visual grading of grain products by machine vision," *Journal of Electronic Imaging*, vol. 24, p. 061116, 11 2015.
- [18] F. Guevara-Hernandez and J. Gomez-Gil, "A machine vision system for classification of wheat and barley grain kernels," *Spanish Journal of Agricultural Research*, vol. 9, p. 672, 09 2011.
- [19] N. Patil, V. Malemath, and R. M. Yadahalli, "Color and texture based identification and classification of food grains using different color models and haralick features," *International Journal on Computer Science and Engineering*, vol. 3, 12 2011.
- [20] M. Rahnemounfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors (Basel, Switzerland)*, vol. 17, 04 2017.
- [21] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," 05 2017, pp. 3626–3633.

## References

- [22] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. Mccool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, p. 1222, 08 2016.
- [23] C. Igathinathane, U. Ulusoy, and L. Pordesimo, "Comparison of particle size distribution of celestite mineral by machine vision volume approach and mechanical sieving," *Powder Technology - POWDER TECHNOL*, vol. 215-16, 01 2012.
- [24] C. Igathinathane, L. Pordesimo, E. Columbus, W. Batchelor, and S. Sokhansanj, "Sievelless particle size distribution analysis of particulate materials through computer vision," *Computers and Electronics in Agriculture - COMPUT ELECTRON AGRIC*, vol. 66, pp. 147–158, 05 2009.
- [25] E. Hamzeloo, M. Massinaei, and N. Mehrshad, "Estimation of particle size distribution on an industrial conveyor belt using image analysis and neural networks," *Powder Technology*, vol. 261, p. 185–190, 07 2014.
- [26] M. Frei and F. Kruis, "Image-based size analysis of agglomerated and partially sintered particles via convolutional neural networks," *Powder Technology*, vol. 360, 10 2019.
- [27] J. Schäfer, P. Schmitt, M. W. Hlawitschka, and H.-J. Bart, "Measuring particle size distributions in multiphase flows using a convolutional neural network," *Chemie Ingenieur Technik*, vol. 91, no. 11, pp. 1688–1695, 2019. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.201900099>
- [28] K. Sharma, M. Gold, C. Zurbrügg, L. Leal-Taixé, and J. Wegner, "Histonet: Predicting size histograms of object instances," 03 2020, pp. 3626–3634.
- [29] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.
- [30] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [31] S. Zhang, L. Wen, Z. Lei, and S. Z. Li, "Refinedet++: Single-shot refinement neural network for object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2020.
- [32] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, "Metaanchor: Learning to detect objects with customized anchors," in *Advances in Neural Information Processing Systems 31*, S. Bengio,

## References

- H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 320–330. [Online]. Available: <http://papers.nips.cc/paper/7315-metaanchor-learning-to-detect-objects-with-customized-anchors.pdf>
- [33] D. Song, Y. Qiao, and A. Corbetta, “Depth driven people counting using deep region proposal network,” 07 2017, pp. 416–421.
- [34] L. Zhang, L. Lin, X. Liang, and K. He, “Is faster R-CNN doing well for pedestrian detection?” in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer, 2016, pp. 443–457. [Online]. Available: [https://doi.org/10.1007/978-3-319-46475-6\\_28](https://doi.org/10.1007/978-3-319-46475-6_28)
- [35] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, “SSH: single stage headless face detector,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 4885–4894. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.522>
- [36] M. Liao, B. Shi, and X. Bai, “Textboxes++: A single-shot oriented scene text detector,” *IEEE Transactions on Image Processing*, vol. PP, 01 2018.
- [37] X. Xiao, Z. Zhou, B. Wang, L. Li, and L. Miao, “Ship detection under complex backgrounds based on accurate rotated anchor boxes from paired semantic segmentation,” *Remote Sensing*, vol. 11, p. 2506, 10 2019.
- [38] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3296–3297.
- [39] 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*. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-10602-1\\_48](http://dx.doi.org/10.1007/978-3-319-10602-1_48)
- [40] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” 2010.
- [41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

## References

- [42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [43] 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.
- [44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

# Paper C

## SieveNet: Estimating the Particle Size Distribution of Kernel Fragments in Whole Plant Corn Silage

Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B.  
Moeslund

This paper has been accepted as a full paper at the *17th International Conference on Computer Vision Theory and Applications (VISAPP)* and will be presented February 2022.

© 2021 SCITEPRESS  
*The layout has been revised.*



## Abstract

*In this paper we present a method for efficiently measuring the particle size distribution of whole plant corn silage with a sieving-based network. Our network, SieveNet, learns to predict the size class of predefined sieves for kernel fragments through a novel sieve-based anchor matching algorithm during training. SieveNet improves inference timings by 40% compared to previous approaches that are based on two-stage recognition networks. Additionally, an estimated Corn Silage Processing score from the network predictions show strong correlations of up to  $0.93 r^2$  against physically sieved samples, improving correlation results by a number of percentage points compared to previous approaches.*

## 1 Introduction

Efficient evaluation of Whole Plant Corn Silage (WPCS) is an important step to determine if the plant is correctly harvested with a forage harvester. One key parameter is the appropriate processing of kernels into smaller fragments. The fragmentation of the corn kernels allows for more efficient and higher quality fodder for dairy cows [1] and is achieved by altering the processing gap in the kernel processor in the harvester. By evaluating the kernel processing a farmer is able to react in the field to suboptimal settings or variation across their field. An efficient evaluation can be beneficial as modern forage harvester are able to harvest multiple tonnes per hour [2]. However, current industry standards are based upon determining the particle size distribution (PSD) of a WPCS sample with manual sieving techniques which require potentially errorsome manual preparation steps. Examples include the Corn Silage Processing Score (CSPS) that measures the percentage of kernel fragments passing a 4.75 mm sieve [1] or the Penn State Particle Separator that determines the distribution over three to four differently sized sieves [3].

Compared to previous similar works on evaluating WPCS our approach is considerably simpler. Previous works have trained two-stage object recognition networks in the form of bounding-box detectors or instance segmentation networks for fine-grain localisation [4, 5]. Then for a set of predictions over a number of images the length of the major axis was compared against the CSPS quality metric. Instead in this work we propose to discard the second stage in the two-stage networks and only adopt an altered Region Proposal Network (RPN). We introduce the network SieveNet that aims to mimic the sieving process that allow for measurements such as CSPS. The network uses a novel anchor matching algorithm during training that allows the network to learn how to classify which sieve size a kernel fragment instance would lie in during sieving. Traditionally, anchors in the RPN are used as dense bounding-box priors of varying sizes computed over the entire

feature map producing object proposals with class-agnostic objectness scores and box refinement deltas. This scheme is altered in SieveNet by defining anchors based on a number of sieving sizes and during training positive anchors are matched using a set of criterion based on sieving. The criterion are:

1. A matched bounding-box anchor should have a diameter smaller than that of the ground truth diameter.
2. The matched bounding-box anchor should be the that which has the smallest difference between the anchor diameter and ground truth diameter.
3. Only a single anchor sieve size can be matched to a ground truth instance.

We adopt the same dataset as the two-stage recognition networks [4, 5] which exhibits a high amount of clutter amongst kernel fragments. An example image from the dataset can be seen in Figure C.1 visualising annotated kernel fragments by a white outline.



**Fig. C.1:** Example of WPCS with annotations of kernel fragments.

The above sieving criteria implemented on the dataset are visualised for a single instance in Figure C.2 highlighting the difference between traditional RPN matching [6] and our novel sieve-based matcher. In both examples a ground truth kernel fragment bounding-box is highlighted by a dashed white outline. In Figure C.2b, during training a positive label is given to the anchors with an Intersection-over-Union (IoU) greater than 0.7, which in this case are marked in green. However, in our approach in Figure C.2c two positive examples are now marked as negatives as their diameter is greater than that

## 2. Related Work

of the ground truth. Additionally, only a single positive match is found which is the first anchor with a smaller diameter. The only requirement we introduce on intersection is that it must be greater than 0, therefore, in theory as long as the three criteria above are met the intersection between anchor and ground truth can be small.

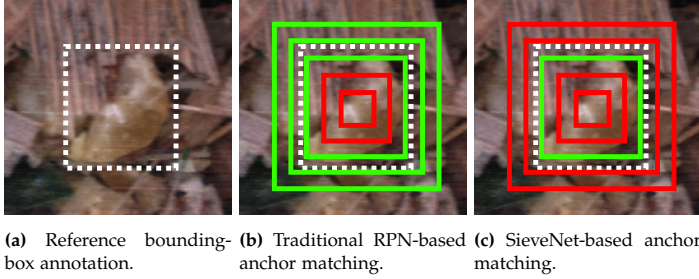


Fig. C.2: Overview of matching strategies between RPN IoU (b) and SieveNet (c).

In comparison to an RPN the SieveNet is simplified in regards to network training as bounding-box regression is not required as we are only interested in the classification of instances into a fixed sieve class. We show in this work that it is possible to train our SieveNet to accurately and efficiently estimate the sieving of WPCS. Finally, compared to previous works we show competitive results in comparison to physically sieved samples at a considerable reduction in inference time.

Our contribution in this work is:

- A novel sieve-based matching algorithm.
- Show that a Region Proposal Network is able to learn to classify a specific bounding-box anchor.
- Improve the speed of kernel fragmentation analysis in WPCS compared to previous methods without compromising CSPS estimation against real-world samples.

## 2 Related Work

Measurement of WPCS through computer vision is limited. Overall there have been two general methodologies; methods that first separate a sample of WPCS such that kernel fragments can easier be localised and methods that analyse images of samples without the need for separation. Within the separation-based approaches, the contours of kernel fragments spread out on a black background were found using maximally stable external regions from which the maximum inscribed circle was compared to determine

CSPS [7]. However, the manual separation steps can be cumbersome to conduct especially for a farmer in the field. Additionally, laboratory equipment is required which can make the process time-consuming and does not allow the farmer to react to their field conditions during the harvest. To address this a number of works have estimated CSPS on non-separated samples of WPCS. Firstly, two-stage object recognition networks in the form of Multi-task Network Cascades and Region-based Fully Convolutional Networks were trained from which CSPS was estimated from instance masks or bounding-boxes and compared against CSPS estimated from annotations [4]. A two-stage Faster R-CNN network was optimised by altering the anchor priors in the RPN by sampling the shapes of training bounding-boxes with k-means clustering [5]. This work also compared model estimated CSPS from the bounding-boxes against a number of physically sieved samples showing a strong correlation over a number of different machine settings. While the works on non-separated samples show good correlation results the networks exhibit a higher range of complexity making them not suitable for an embedded system where processing power is limited. Additionally, the two-stage pipeline of region proposals followed by box refinement may be superfluous as the final predictions end up being compared against a single CSPS threshold.

In other domains a number of works attempt to measure the size distribution of objects. These include determining the PSD of overlapping iron ore using hand-crafted shape and size features [8]. A U-Net based semantic segmentation network has also been used to localise iron ore pellets [9]. The grain size of beach pebbles were estimated using a Mask R-CNN showing positive correlation when mapping the size results against measured samples [10]. A novel multi-task network architecture, HistoNet, has been used to predict a count map and a histogram without the need for fine-grain localisation of objects in cluttered scenes [11]. This work aims to move away from the complex pipeline found in object recognition networks and show impressive results compared to a Mask R-CNN. A significant amount of the training data is simulated which is possible due to the lower amount of variation in colour and texture in the images.

Other examples exist in literature of the RPN being utilised to localise objects without using the second half of the two-stage pipeline. Firstly, the generalisation of the RPN has been analysed on a number of benchmark datasets for multispectral person detection showing that the network could produce good quality predictions [12]. An RPN with a custom backbone architecture has been used to localise organs in 3D images from CT scans [13]. The authors also included multi-class scores and together with box refinement and fusion of 3D feature data provided accurate results.

### 3 Methodology

SieveNet is built upon the RPN introduced in Faster R-CNN [6] with a ResNet50 [14] backbone within the Pytorch Detectron2 [15] framework. The aim of SieveNet is to efficiently determine the PSD within an RGB image given user-defined priors giving sieve sizes. Additionally, the network follows the supervised-learning mantra and therefore requires annotated instances of relevant objects in bounding-box format. The novel matching algorithm between anchors and ground truth boxes moves away from a purely Intersection-over-Union (IoU) criteria but rather matches based upon how an instance would be sieved. For example, a kernel fragment with a diameter of  $x$  would pass sieves which have a diameter greater than  $x$  but not those which are smaller. Therefore, our matching algorithm finds for each ground truth instance the anchor diameter that matches the sieving criterion defined earlier.

#### 3.1 Dataset

We adopt the dataset for training SieveNet first presented in the works for performing object recognition for kernel fragmentation with two-stage networks [4]. The dataset contains a total of 2438 images containing 11601 annotated kernel fragments split over training, validation and test sets.

#### 3.2 Anchor Matching

The matching of anchors as positive or negative samples during training in the traditional RPN is based upon an IoU approach between the anchor and ground truth boxes at each sliding window location. If a given anchor has an IoU above a certain threshold with a ground truth box the anchor is labelled as a potential positive sample, anchors with an IoU below another threshold are labelled as negatives and finally the anchor boxes with an IoU between the two threshold are given an ignore label. Typical threshold values defined in the original introduction of the RPN in Faster R-CNN [6] are 0.7 for positives and 0.3 for negatives, however, these can be altered to a given use-case. Finally, a distribution of positive and negative boxes are sampled for each image during training with the network learning the class-agnostic probability between object and background. As mentioned, this matching strategy is not sufficient for efficiently estimating PSD given a sieving criteria as positive matches can include boxes where either the anchor or ground truth has the larger diameter. Additionally, depending on the chosen anchor prior multiple anchors can be labelled as positives as long as each IoU is greater than the chosen threshold.

Our approach to anchor matching is first to define anchor shapes that match a potential sieving system which could be used to estimate CSPS. A total of five anchor sieves are chosen ranging from 1 mm to 9 mm in increments of 2 mm. Due to the constant distance between camera and samples when capturing images in the dataset this equates to pixel ranges between 20 and 180 at increments of 20. An overview of the sieve matching is covered in Algorithm 1. First, for a set of ground truth boxes the IoU is calculated with the anchors at each position in the feature map. Next, for all ground truth boxes and anchor boxes the diameters of each box is determined, for ground truth boxes the diameter is taken as the larger of the two axes. Then for each coordinate in the feature map with an IoU greater than zero the five anchor diameters are compared to the given ground truth diameter and the anchor diameters that are smaller are given a positive label. Anchor diameters that are greater represent sieves where the instance would pass are given a negative label. Once completed for all ground truth boxes, at each coordinate with multiple positives the positive anchors that do not have the smallest diameter and are set to negatives. At this point at each coordinate with an initial IoU greater than zero the correct sieve-based anchor is now matched. Finally, Non-Maximum Suppression (NMS) is applied for positive anchor labels at a threshold of 0.9 where anchors that overlap greatest with the ground truth are prioritised for training samples.

---

**Algorithm 1:** Anchor matching algorithm for SieveNet.

---

**Function:** SIEVEMATCHER(*gtboxes*, *anchors*)  
 Calculate IoU(*gtboxes*, *anchors*)  
 Calculate diameters of GT boxes  
 Calculate diameters of anchor boxes  
**for** *each coordinate with IoU > 0* **do**  
   **if** *Anchor diameter < GT diameter* **then**  
     anchor label = 0  
   **else**  
     anchor label = 1  
**for** *Coordinates with multiple label == 1* **do**  
   Find smallest anchor diameter label == 1  
   Anchor labels where not smallest = 0  
   Apply NMS at threshold 0.9 for positive anchors  
**return:** Anchor labels ;  
**end function**

---

The number of positive samples is significantly different when adopting the matching approach compared to the IoU matching. In our networks we do not take into account an IoU threshold and allow matches to be set as

### 3. Methodology

long as the IoU is greater than zero. This approach mimics sieving better as a correctly sieved object may be considerably larger than the sieve/anchor resulting in a poor IoU. An alternative to our matching method is to adopt the Intersection-over-Area (IoA) metric in the RPN matching step. In IoA the overlap is defined as the area of the intersection over the area of the anchor box. A potentially more relevant metric is our sieve matching only uses cases where the anchor is the smaller of the two boxes. Table C.1 shows the difference in the number of positive samples for the images in the training set before applying NMS to find the highest quality matches. A considerably larger amount of positive examples exist when using the IoA metric compared to IoU in the RPN matching equating to on average around 88 samples compared to 10. This is likely due to smaller anchors encapsulated by a ground truth scoring 1.0 instead of a potentially much lower score with IoU. Finally, our approach finds  $2.75\times$  more positives than the IoU approach despite only allowing a single anchor match at each location, however, we do match positives independent of any intersection based metric.

Matcher	Positive Samples
RPN IoU	14026
RPN IoA	122966
SieveNet	38652

**Table C.1:** Number of positive samples for the different matching methods for all images in the training set.

Finally, we perform our sieve matching at a stride of 1 in the feature map. Other options exist, however, care should be taken dependent on the chosen backbone architecture. In our case, with ResNet50, the backbone down samples the input image throughout the network by a number of pooling and striding operations resulting in a feature map four times smaller. Therefore, when applying anchor matching at a stride of 1 this equates to a stride of 8 pixels in the input. For SieveNet with ResNet50 this difference is negligible but with a different architecture or changing the stride in the feature map may result in lower effectiveness in the matching step.

During inference the anchor matching step is naturally not included. Instead, the SieveNet uses a sliding window at the stride of 1 over the feature map and predict the probability of each anchor matching with a kernel fragment. Then predictions are thresholded based upon a confidence score and NMS thresholds predictions at an IoU of 0.05 leaving the final sieved predictions.

### 3.3 Model Training

The SieveNet with the anchor matching strategy presented in the previous section are trained for a total of 25000 iterations using stochastic gradient descent with a base learning rate of 0.025 and a batch size of four. Images are rescaled such that the shortest axis is 600 pixels and horizontal flipping augmentation is applied to double the amount of images. The training and inference of the models used for the results presented in the next section are done on an NVIDIA Titan XP GPU. During evaluation of the networks we take the given network iteration with the lowest validation loss.

## 4 Results

In this section we present results from SieveNet models. This includes studies comparing both within model SieveNet variants are against an RPN with the classic matching algorithm. To make the results comparable between SieveNet and the RPNs we also remove bounding-box refinement from the RPNs. We present correlation results for models based upon the dataset of physically sieved samples for CSPS from two harvest weeks presented in [5] and compare against the Faster R-CNN models from the same work. The data for the samples includes image sets and CSPS scores for a number of harvest runs containing machine setting altering the kernel fragmentation. For an image set we run our models over all images and estimate the CSPS by determining the percentage of predictions that pass the 5 mm anchor. When evaluating the models we present results with the Pearson Correlation Coefficient (PCC),  $r^2$  coefficient of determination and the Root Mean Square Error (RMSE) comparing estimated model CSPS and physically sieved CSPS.

### 4.1 Matching Strategy

In Figures C.3a and C.3b example predictions from the same image are shown for RPN trained with the IoU and IoA respectively, where in Figures C.3c predictions for SieveNet are shown. The example predictions in Figure C.3b show the limitations of using an IoA based approach with RPN original matching approach. Here, any anchors that are within the bounds of a ground truth measure as 1.0 resulting in many small anchors being matched per ground truth. Additionally, as no bounding-box refinement is learnt NMS cannot be used to discard multiple anchors covering the same instance. Both RPN with the IoU metric and SieveNet show visually promising results appearing to match anchor boxes well with kernel fragment instances.

Table C.2 shows correlation results at three different confidence thresholds for each of the matching methods. Each approach show strong correlation



#### 4. Results

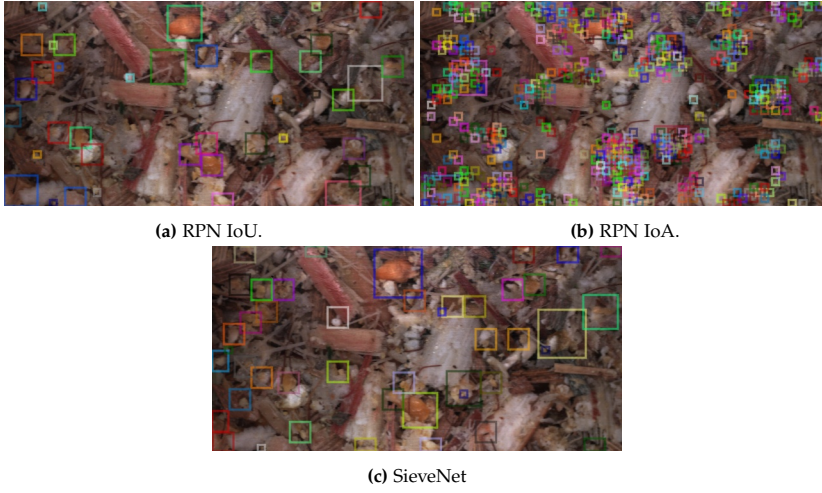


Fig. C.3: Example predictions from models trained on different matching strategies.

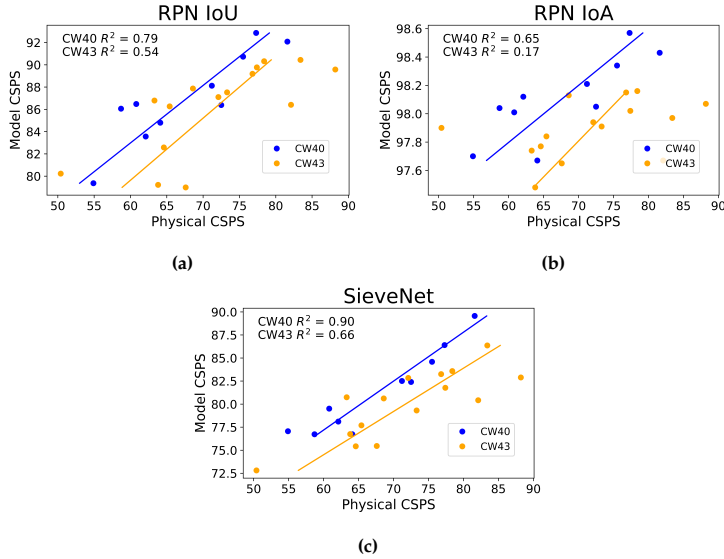
scores, the RPN methods adopting an IoU threshold shows similar results in terms of PCC and  $r^2$  compared to previous Faster R-CNN approaches. However, correlation scores decrease when adopting IoA especially for the CW43 dataset. SieveNet improves the results for both harvest weeks increasing both PCC and  $r^2$  by a number of percentage points. For each of the approaches it can also be seen that the confidence threshold has an effect on CSPS correlation. For SieveNet there appears to be a good trade-off at confidence 0.25 show strong results for both weeks.

When evaluating the correlation results for both harvest weeks together SieveNet does show a slight decrease compared to Faster R-CNN as shown in Table C.3.

Figure C.4 shows a scatter plot of physical CSPS measured for each sample compared to estimated model CSPS for the image sets from CW40 and CW43 which are also shown in Table C.2. We see the positive correlation for all three approaches, especially strong at the values from SieveNet with well aligned points.

Model	CW40		CW43			
	PCC	$r^2$	RMSE	PCC	$r^2$	RMSE
FRCNN Baseline [5]	0.68	0.46	8.12	0.64	0.41	17.09
FRCNN 2a (conf: 0.5) [5]	0.84	0.70	<b>5.39</b>	0.63	0.40	8.89
FRCNN 2a (conf 0.25) [5]	0.90	0.80	8.90	0.66	0.44	7.64
FRCNN 2a (conf 0.05) [5]	0.91	0.84	18.87	0.77	0.59	16.23
RPN IoU (conf 0.5)	0.88	0.78	18.93	0.69	0.49	14.70
RPN IoU (conf 0.25)	0.89	0.79	19.90	0.74	0.54	16.09
RPN IoU (conf 0.05)	0.89	0.79	21.49	0.75	0.56	17.78
RPN IoA (conf 0.5)	0.82	0.67	31.38	0.47	0.22	27.80
RPN IoA (conf 0.25)	0.81	0.65	31.35	0.41	0.17	27.81
RPN IoA (conf 0.05)	0.80	0.64	31.34	0.38	0.15	27.73
SieveNet (conf 0.5)	<b>0.96</b>	<b>0.93</b>	10.12	0.74	0.54	<b>7.50</b>
SieveNet (conf 0.25)	0.95	0.90	14.27	<b>0.81</b>	<b>0.66</b>	10.70
SieveNet (conf 0.05)	0.85	0.73	31.11	0.44	0.19	27.42

**Table C.2:** Correlation results for previous works with Faster R-CNN (FRCNN) and our three networks with different matching strategies for two separate harvest weeks.



**Fig. C.4:** Correlation plots for the three matching strategies.

Finally, we see that in Table C.4 that SieveNet improves the inference time by almost 40% compared to Faster R-CNN when evaluating an inference image on an NVIDIA Titan XP GPU.

## 4. Results

Model	CW40+ CW43		
	PCC	$r^2$	RMSE
FRCNN Baseline [5]	0.53	0.28	14.2
FRCNN 2a (conf: 0.5) [5]	0.64	0.40	<b>7.69</b>
FRCNN 2a (conf: 0.25) [5]	0.71	0.51	8.17
FRCNN 2a (conf: 0.05) [5]	<b>0.81</b>	<b>0.66</b>	17.34
RPN IoU (conf: 0.5)	0.70	0.49	16.50
RPN IoU (conf: 0.25)	0.75	0.56	17.71
RPN IoU (conf: 0.05)	0.76	0.58	19.35
RPN IoA (conf: 0.5)	0.43	0.19	29.29
RPN IoA (conf: 0.25)	0.43	0.18	29.28
RPN IoA (conf: 0.05)	0.30	0.09	29.23
SieveNet (conf: 0.5)	0.75	0.56	8.65
SieveNet (conf: 0.25)	0.80	0.64	12.27
SieveNet (conf: 0.05)	0.48	0.23	27.23

**Table C.3:** Correlation results for previous works with Faster R-CNN (FRCNN) and our three networks for a combined correlation over both harvest weeks.

Model	Inference Time (ms)
FRCNN 2a [5]	51.1
SieveNet	<b>34.1</b>

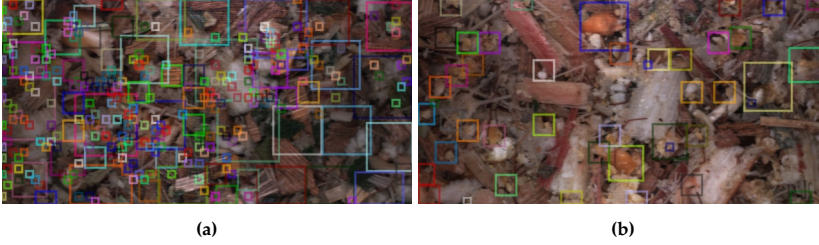
**Table C.4:** Timings for networks on an NVIDIA Titan XP GPU.

### 4.2 Number of Anchors

For estimating the physical characteristics of the harvested crop, our aim was to predict the CSPS of a sample. For CSPS only the single sieve of 4.75 mm is required but in practice is typically done with multiple different sizes. In this section we evaluate training a SieveNet with two anchors, one for a 4.75 mm sieve and a smaller anchor at 1 mm capturing particles that pass the CSPS threshold anchor.

Figure C.5 shows an example prediction from SieveNet with either two or five anchors. At the former in Figure C.5a the restriction of not adopting box refinement allowing for further NMS is clear. Smaller fragments passing the larger 4.75 mm sieve that are more than double the size of the smallest anchor have multiple predictions, similar that in when using the IoA metric in the RPN. This effect is counteracted when training with more anchor sieves with a consistent increment in diameter as instances are never 100% greater than an associated sieve.

In Table C.5 the correlation results together with previously presented



**Fig. C.5:** Two anchors for sieve sizes 1 mm and 4.75 mm. Five anchors for size sizes between 1 mm and 9 mm with 2 mm increments.

models are shown for SieveNet with two anchors. The results highlight what was visualised in the image with poor correlation, especially at CW43.

Model	CW40			CW43		
	PCC	$r^2$	RMSE	PCC	$r^2$	RMSE
FRCNN Baseline [5]	0.68	0.46	8.12	0.64	0.41	17.09
FRCNN 2a (conf: 0.5) [5]	0.84	0.70	<b>5.39</b>	0.63	0.40	8.89
FRCNN 2a (conf 0.25) [5]	0.90	0.80	8.90	0.66	0.44	7.64
FRCNN 2a (conf 0.05) [5]	0.91	0.84	18.87	0.77	0.59	16.23
RPN IoU (conf 0.25)	0.89	0.79	19.90	0.74	0.54	16.09
RPN IoA (conf 0.25)	0.81	0.65	31.35	0.41	0.17	27.81
SieveNet (conf 0.25)	<b>0.95</b>	<b>0.90</b>	14.27	<b>0.81</b>	<b>0.66</b>	10.70
SieveNet two anchors (conf 0.25)	0.38	0.15	29.42	-0.17	0.3	23.32

**Table C.5:** Correlation for the two harvest weeks with previous results and additionally SieveNet with two anchors.

## 5 Conclusion

In this work we present SieveNet, a network able to efficiently monitor WPCS in RGB images captured directly from a forage harvester. We show that localisation of kernel fragments is viable only with an RPN-based architecture reducing the complexity compared to previous approaches based on two-stage recognition networks. Additionally, we introduce an anchor matching algorithm giving the ability to train networks to classify kernel fragments into predefined sieve sizes. These predictions allow for estimation of CSPS with a strong correlation against physical samples. We believe SieveNet can be extended to other domains where the PSD is also of interest, such as agglomerates or medical imaging, given a definition of appropriate sieve-based anchors.

## References

- [1] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," *In: Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [2] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [3] J. Heinrichs and M. J. Coleen, "Penn state particle separator," May 2013. [Online]. Available: <https://extension.psu.edu/penn-state-particle-separator>(accessedon24July2018)
- [4] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [5] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Anchor tuning in faster r-cnn for measuring corn silage physical characteristics," *Computers and Electronics in Agriculture*, vol. 188, p. 106344, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169921003616>
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>
- [7] J. L. Drewry, B. D. Luck, R. M. Willett, E. M. Rocha, and J. D. Harmon, "Predicting kernel processing score of harvested and processed corn silage via image processing techniques," *Computers and Electronics in Agriculture*, vol. 160, pp. 144 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918311955>
- [8] T. Andersson, M. J. Thurley, and O. Marklund, "Visibility classification of pellets in piles for sizing without overlapped particle error," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, 2007, pp. 508–514.

## References

- [9] J. Duan, X. Liu, X. Wu, and C. Mao, "Detection and segmentation of iron ore green pellets in images using lightweight u-net deep learning network," *Neural Computing and Applications*, vol. 32, pp. 5575–5790, 2020.
- [10] A. Soloy, I. Turki, M. Fournier, S. Costa, B. Peuziat, and N. Lecoq, "A deep learning-based method for quantifying and mapping the grain size on pebble beaches," *Remote Sensing*, vol. 12, no. 21, 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/21/3659>
- [11] K. Sharma, M. Gold, C. Zurbrügg, L. Leal-Taixé, and J. Wegner, "Histonet: Predicting size histograms of object instances," 03 2020, pp. 3626–3634.
- [12] K. Fritz, D. Koenig, U. Klauck, and M. Teutsch, "Generalization ability of region proposal networks for multispectral person detection," *Automatic Target Recognition XXIX*, May 2019. [Online]. Available: <http://dx.doi.org/10.1117/12.2520705>
- [13] X. Xu, F. Zhou, B. Liu, D. Fu, and X. Bai, "Efficient multiple organ localization in ct image using 3d region proposal network," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1885–1898, 2019.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 06 2016, pp. 770–778.
- [15] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.

## **Part III**

# **Understanding Neural Networks**





# Paper D

## Evaluation of Model Selection for Kernel Fragment Recognition in Corn Silage

Christoffer Bøgelund Rasmussen and Thomas B. Moeslund

The paper has been published in  
*ICLR 2020 Workshop on Computer Vision for Agriculture (CV4A) - Virtual, 2020.*

© 2020

*The layout has been revised.*

# Abstract

*Model selection when designing deep learning systems for specific use-cases can be a challenging task as many options exist and it can be difficult to know the trade-off between them. Therefore, we investigate a number of state of the art CNN models for the task of measuring kernel fragmentation in harvested corn silage. The models are evaluated across a number of feature extractors and image sizes in order to determine optimal model design choices based upon the trade-off between model complexity, accuracy and speed. We show that accuracy improvements can be made with more complex meta-architectures and speed can be optimised by decreasing the image size with only slight losses in accuracy. Additionally, we show improvements in Average Precision at an Intersection over Union of 0.5 of up to 20 percentage points while also decreasing inference time in comparison to previously published work. This result for better model selection enables opportunities for creating systems that can aid farmers in improving their silage quality while harvesting.*

## 1 Introduction

Computer vision systems for quality inspection are widespread throughout agriculture and many other industries. Deep learning has become the driving force in many applications largely due to advantages such as potentially high accuracy and ease of use due to the large number of open source libraries. The common methodology for training the networks is either to adapt an open-source network or for an author to design their own network. However, it can be difficult to choose which network is best for a specific task as it often comes with a trade-off between complexity, accuracy and speed. Therefore, in this work our contribution is showing a systematic approach is create an overview over the trade-off for a specific agricultural task of corn kernel fragment recognition from corn silage harvested from a forage harvester. In corn silage kernels must be cracked sufficiently such that when used as fodder for dairy cows the starch content is easily ingested and milk yield can be optimised [1]. An recognition system for high quality can help farmers use their machine optimally, avoiding both quality decreasing by up to 25% and inefficient usage of diesel fuel [2]. Furthermore, such systems can help solve the potential food crisis as the population is expected to reach 9.1 billion in 2050 [3].

This work extends upon that done in [4] where it was shown that kernel fragment shape and size characteristics could be measured with Convolutional Neural Networks (CNNs) for bounding-box detection and instance segmentation, however, only a single form of each was trained and it is unknown if these architectures are optimal. In [5] the trade-off between speed and accuracy was explored for CNN-based object detectors. Whilst compre-

hensive and useful as the open-source implementations are available through TensorFlow object detection API, networks are trained and evaluated on the large COCO benchmark dataset [6] and it is not as clear what the trade-off is for a specific use-case on a smaller scale like kernel fragmentation. We provide an overview of the trade-off for the kernel recognition by training variants of three meta-architectures of increasing complexity with the API from [5] and explore different feature extractors and input image resolutions. This allows us to show an approach to determine optimal model design choices for CNN-based kernel fragment recognition.

## 2 Data

The data used to train and test the networks are the same as that used in [4] and consist of RGB images of silage taken post-harvest. Typically, kernel processing evaluation requires the separation of kernels and stover (leaves and stalks) either through manual means as in [7, 8] followed by sieving measurements or sieving estimation with image processing [9]. However, the manual separation step can be cumbersome making it problematic for a farmer whilst harvesting. Therefore, in [4] images and annotations were collected of non-separated corn silage for a direct measurement.

The dataset consists of a total of 2043 images with 11601 kernel fragment annotations. A notable difference in this work compared to [4] is a validation set is added to combat overfitting whilst training by evaluating a model variant with the lowest validation loss. In [4] the data was split 60% for training and 40% for testing, here we keep the same training set but evenly split the original test set such that validation and test cover 20% each. For the variation of image sizes when training and testing models images are resized from the original images dimensions of  $640 \times 1280$  to either  $600 \times 1200$ ,  $400 \times 730$  or  $200 \times 365$  using bilinear interpolation.

## 3 CNN Meta-Architectures

The TensorFlow object detection API provides a number of options for meta-architectures and includes pre-trained models with different backbone feature extractors and hyperparameters. Hyperparameters for the training of our models remained unchanged to the configurations files provided in the API, apart from the learning rate being decreased by a factor of 10 as only fine-tuning is performed. Networks are trained using TensorFlow 1.13.1 on a machine containing an NVIDIA Titan XP and GTX 1080Ti.

The first meta-architecture adopted is the Single Shot Multibox Detector (SSD) and is an efficient single-stage bounding-box detector. SSD has a

competitive accuracy whilst running much faster than other more complex networks. For the varying complexity of feature extraction within SSD we adopt MobileNetv1 [10], MobileNetv2 [11] and InceptionV2 [12]. Next, we train Faster R-CNN, a two-stage bounding-box detector that utilises the Region Proposal Network (RPN) to produce candidate proposals whose boxes are regressed and classified. For Faster R-CNN we train variants with Inceptionv2, ResNet50 and ResNet101 from [13]. Lastly and most complex is the instance segmentation network Mask R-CNN [14]. The network is an extension of Faster R-CNN but with the added ability of producing masks for prediction. As the RPN is also part of Mask R-CNN the network is also able to output bounding-boxes, thus both forms will be evaluated. The feature extractors trained for Mask R-CNN are also Inceptionv2, ResNet50 and ResNet101.

## 4 Results

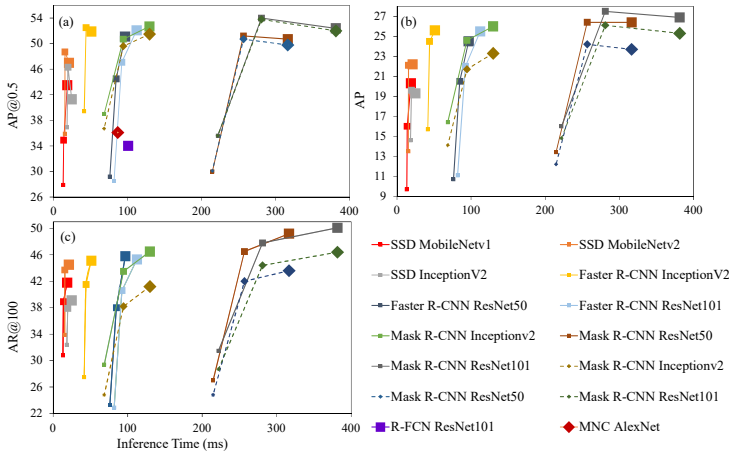
The results in Table D.1 are based upon a subset of the COCO metrics where the models with bounding-box predictions can be seen in first section and segmentation models in the second section. Additionally, we show the AP@0.5 results from [4] for R-FCN [15] with ResNet101 and the MNC [16] with AlexNet [17]. As mentioned in Section D.2, we altered the test set such that a validation set is also available. Therefore, the results are not calculated on the exact same images as in [4] but we argue that the new test set is large enough such that the results are comparable.

The results in Table D.1 are visualised in Figure D.1 where we show the AP@0.5 in (a), AP in (b) and AR@100 in (c) all against the inference time of the models. Firstly, we see a significant improvement in the AP@0.5 in comparison to the R-FCN model from [4] in addition to a decrease in inference time for all SSD variants and some of the Faster R-CNNs and Mask R-CNNs. The models trained in this work have an AP@0.5 of around 20 percentage points higher, while running at up to  $5\text{-}8\times$  faster for bounding-boxes. However, the segmentation variants proved to be slower than previous with only the Mask R-CNN Inceptionv2 at image size  $200\times 365$  running  $1.27\times$  faster and improving AP@0.5 by 0.6 percentage points in comparison to the MNC model from [4]. However, improvements of up to 17.7 percentage points are seen for more complex models but at a cost of increased inference time.

Comparing the varying meta-architecture complexity we see that there is a slight gain in the metrics when evaluating bounding-box outputs. However, this comes at a cost of inference time, especially between Faster R-CNN and Mask R-CNN. Within each meta-architecture we see slight differences between feature extractors. At  $600\times 1200$  AP for SSD improves by 9.4% from MobileNetv1 to MobileNetv2 but falls for Inceptionv2, Faster R-CNN in-

creases by 4.5% from Inceptionv2 to ResNet101 and Mask R-CNN by 3.5% from Inceptionv2 to ResNet101. This shows that less is gained spending time on determining the optimal architecture for feature extraction in comparison to choosing the meta-architecture. This is in contrast to the findings in [5] where large improvements could be made, for example, Faster R-CNN had a 70% increase in AP on the MS COCO test set over the evaluated feature extractors. Finally, we do see improvements in the metrics when increasing the image size from  $200 \times 365$  to  $400 \times 730$ , but not as much when between  $400 \times 730$  and  $600 \times 1200$ . Additionally, a significant increase in inference time is seen for most meta-architectures when the image size is at the largest.

Lastly, an example image with predictions from the best performing model with respect to AP and AP@0.5 can be seen in Figure D.2.



**Fig. D.1:** Results from the model variants for AP@0.5 (a), AP (b) and AR@100 (c) against inference time on an NVIDIA Titan XP. Models producing bounding-box outputs are shown with a solid line and square points and segmentation outputs are shown with a dashed line and diamond points. The increase in image size is shown by an increase in the size of the respective points.



**Fig. D.2:** Left: Mask R-CNN ResNet101 (400x730) predictions. Right: Ground truth annotations.

## 5 Conclusions

In this work we have shown a systematic approach to train object recognition networks towards the task of kernel fragment recognition in corn silage whilst providing an overview of the trade-off in complexity, accuracy and speed. We show that slight improvements in AP and AR can be made by adopting more complex meta-architectures but at a larger cost of inference time. For all models the gain in AP and AR from a small to a medium image size was considerable, however, was minimal or worse when increasing onwards to a larger size. Minimal improvements could be made when altering the feature extractor for each meta-architecture, a contrast to findings on COCO in [5] We propose that this approach can be transferred to other similar domains where training data can be sparse in order select an appropriate model and speculate that these design choices for our models could be directly transferred to tasks with similarities in images, such as high amounts of clutter and occlusion. The improvements in kernel fragment recognition through better model selection open possibilities for a more efficient and robust system for farmers to obtain improved yields.

**Table D.1:** Results of the models on the test set. The bounding-box outputs are evaluated are shown in the first section followed by the segmentation outputs.

MODEL	IMAGE SIZE	AP	AP@0.5	AR@100	INFERENCE TIME (ms)
R-FCN ResNet101 [4]	600×1200	NA	34.0	NA	101.0
	600×1200	20.3	43.5	41.8	18.8
	400×730	16.0	34.9	38.9	13.8
	200×365	9.7	27.9	30.8	<b>13.2</b>
SSD MobileNetV1	600×1200	22.2	47.0	44.5	21.1
	400×730	22.1	48.7	43.7	15.6
	200×365	13.5	35.9	33.9	15.4
	600×1200	19.3	41.3	39.1	24.8
SSD MobileNetV2	400×730	19.6	46.3	37.9	19.6
	200×365	14.6	36.9	32.4	18.3
	600×1200	25.6	51.9	45.1	51.1
	400×730	24.5	52.5	41.5	44.1
SSD InceptionV2	200×365	15.7	39.4	27.5	41.6
	600×1200	24.5	51.1	45.8	96.8
	400×730	20.5	44.5	38.0	84.8
	200×365	10.7	29.2	23.3	76.2
Faster R-CNN InceptionV2	600×1200	25.5	52.1	45.3	112.4
	400×730	22.0	47.1	40.6	92.4
	200×365	11.1	28.5	22.9	81.9
	600×1200	26.0	52.7	46.5	129.8
Faster R-CNN ResNet50	400×730	24.6	50.7	43.5	94.5
	200×365	16.4	39.0	29.4	68.5
	600×1200	26.4	50.7	49.2	316.6
	400×730	26.4	51.2	46.5	256.8
Faster R-CNN ResNet101	200×365	13.4	30.0	27.0	214.7
	600×1200	26.9	52.4	<b>50.1</b>	381.5
	400×730	<b>27.5</b>	<b>54.0</b>	47.8	281.1
	200×365	16.0	35.6	34.5	222.0
<hr/>					
MNC AlexNet [4]	600×1200	NA	36.1	NA	87.0
Mask R-CNN InceptionV2	600×1200	23.3	51.5	41.2	129.8
	400×730	21.7	49.6	38.2	94.5
	200×365	14.1	36.7	24.8	<b>68.5</b>
	600×1200	23.7	49.8	43.6	316.6
Mask R-CNN ResNet50	400×730	24.2	50.7	42.0	256.8
	200×365	12.2	30.1	24.8	214.7
	600×1200	25.3	52.0	<b>46.4</b>	381.5
	400×730	<b>26.1</b>	<b>53.8</b>	44.4	281.1
Mask R-CNN ResNet101	200×365	14.8	35.6	28.7	222.0



## References

- [1] L. Johnson, J. Harrison, D. Davidson, W. Mahanna, and K. Shinnors, "Corn silage management: Effects of hybrid, chop length, and mechanical processing on digestion and energy content," *Journal of dairy science*, vol. 86, pp. 208–31, 02 2003.
- [2] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [3] FAO, "How to Feed the World 2050," [http://www.fao.org/fileadmin/templates/wsfs/docs/expert\\_paper/How\\_to\\_Feed\\_the\\_World\\_in\\_2050.pdf](http://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf) (accessed February 10, 2020), 2009.
- [4] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3296–3297.
- [6] 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*. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-10602-1\\_48](http://dx.doi.org/10.1007/978-3-319-10602-1_48)
- [7] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," In: *Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [8] Penn State Extension, "Penn State Particle Separator," <https://extension.psu.edu/penn-state-particle-separator> (accessed February 10, 2020), 2016.
- [9] J. L. Drewry, B. D. Luck, R. M. Willett, E. M. Rocha, and J. D. Harmon, "Predicting kernel processing score of harvested and processed corn silage via image processing techniques," *Computers and Electronics in Agriculture*, vol. 160, pp. 144 – 152, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918311955>

- [10] 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," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [11] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 4510–4520. [Online]. Available: [http://openaccess.thecvf.com/content\\_cvpr\\_2018/html/Sandler\\_MobileNetV2\\_Inverted\\_Residuals\\_CVPR\\_2018\\_paper.html](http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html)
- [12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
- [14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.
- [15] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 379–387. [Online]. Available: <http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>
- [16] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 3150–3158. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.343>
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

# Paper E

## The Challenge of Data Annotation in Deep Learning – A Case Study on Whole Plant Corn Silage

Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B.  
Moeslund

Submitted to *MDPI Sensors* 2021.

© 2021 Christoffer Bøgelund Rasmussen, Kristian Kirk and Thomas B. Moeslund  
*The layout has been revised.*

# Abstract

*Recent advances in computer vision are primarily driven by the usage of deep learning which is known to require large amounts of data and creating datasets for this purpose is not a trivial task. Larger benchmark datasets often have detailed processes with multiple stages and users with different roles during annotation. However, this can be difficult to implement in smaller projects where resources can be limited. Therefore, in this work we present our processes for creating an image dataset for kernel fragmentation and stover overlengths in Whole Plant Corn Silage. This includes the guidelines for annotating object instances in respective classes and statistics of gathered annotations. Given the challenging image conditions, where objects are present in large amounts of occlusion and clutter, the datasets appear appropriate for training models, however, we experience annotator inconsistency which can hamper evaluation. Based upon this we argue the importance of having an evaluation form independent of the manual annotation where we evaluate our models with physically based sieving metrics. Additionally, instead of the traditional time-consuming manual annotation approach we evaluate Semi-Supervised Learning as an alternative showing competitive results while requiring fewer annotations.*

## 1 Introduction

Monitoring the harvesting of Whole Plant Corn Silage (WPCS) with a forage harvester can enable a farmer to react to varying conditions by altering key settings in their machine in order to maximise quality. Current approaches used by farmers are mostly based on manual sieving of samples which give information on the particle size distribution. However, recent works [1, 2] have shown the promise of using deep learning in the form of Convolutional Neural Networks (CNNs) for automatic object recognition in samples taken directly from the machine. These methods have minimal manual steps allowing farmers to efficiently react in the field. However, the usage of CNNs introduces challenges in creating image datasets as it is widely known that models require large amounts of annotated data to train [3]. Large datasets such as ImageNet [4] and COCO [5] have been one of the key reasons for the progression in computer vision over the past decade. Quality and consistency of the annotations is key and often this is acquired through well defined multi-stage processes including team members who take on different roles. Naturally, this can be a time-consuming and expensive process. Alternative or additional methods can also be used to speed up the manual process, including approaches such as transfer learning, weak supervision, or Semi-Supervised Learning (SSL) [6].

The quality of the harvested crop is highly dependent on farmers using correct machine settings for their harvester to react to their field condi-

tions [7]. Two of the key settings are the Processor Gap (PG) and Theoretical Length of Cut (TLOC), that primary effect the fragmentation of kernels and chopping of stover particles respectively. The PG is the gap between rotating processor rolls that compresses and cracks kernels into fragments. The TLOC is controlled by the speed of a rotating knife drum, where a higher speed chops the plant into smaller particles. In Figure E.1 examples from our two forms of datasets are shown. Figure E.1(a) shows an example of kernel fragment annotations. In this case our aim is to create an annotated dataset containing instances of kernel fragments such that we can train a network to perform object recognition and thereby estimate the quality across images. For quality we estimate the industry standard metric Corn Silage Processing Score (CSPS) [8] which gives a measurement of the percentage of kernel fragments passing through a 4.75 mm sieve. A higher CSPS indicates higher quality since the kernels are easier to digest when the WPCS is used as fodder for dairy cows. Figure E.1(b) shows annotations of stover overlength annotations. For kernel fragments the aim was to annotate and predict all instances, however, this task was deemed to be too demanding for stover particles as all remaining instances would have to be marked. Therefore, we only annotated particles marked as overlengths which are classified based on how the WPCS was harvested. Farmers can have different strategies for the chopping of stover particles given their requirements. For example, longer particles can promote cud chewing but shorter particles can be easier to pack in a silo [9]. Therefore, we annotate such that we can measure a dynamic overlength given the farmer's chosen TLOC. This overlength definition is  $1.5 \times \text{TLOC}$ . The WPCS in Figure E.1(b) is harvested with a TLOC of 4 mm and therefore particles greater than 6 mm are annotated. Additionally for stover annotations we annotated four classes covering different parts of the plant. Figure E.1 shows that for both datasets the instances are challenging for both a network to predict but also for annotators to annotate due to the high amounts of clutter between particles.

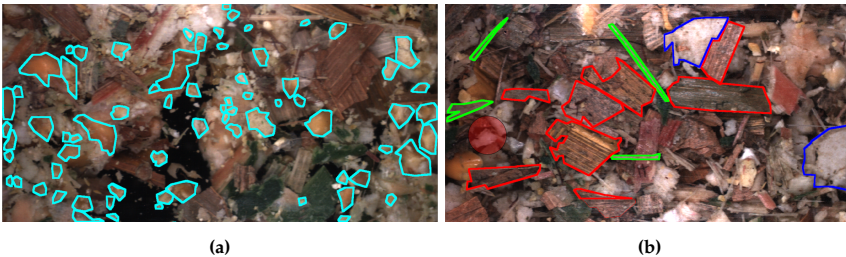


Fig. E.1: Examples annotations of kernel fragments (a) and for stover overlengths in (b).

While highly-defined processes can lead to a high quality dataset, it can be an expense that is not available in all projects, especially in the early phases.

## 2. Related Work

This has been the case for our datasets for WPCS which have been used in a number of works [1, 2, 10], however, have shown to produce promising results. Therefore, in this work we investigate the challenges of data annotation for deep learning. This includes presenting our processes for creating an WPCS image dataset with annotated object instances through manual annotation. We show our guidelines for annotating datasets leading to supervised learning with CNNs. The resulting datasets and models show that the methodology is viable however as the datasets scale to larger sizes through multiple annotators the consistency falters. Annotator disagreement is a common challenge which can be addressed with well-defined processes [3], however, this is costly to create and manage. Alternatively, the field of SSL aims to take advantage of more efficiently gathered higher-level or noisy input to train models [6] which we evaluate for our purpose. While extensive literature exists for the process of creating datasets for larger benchmarks, it is limited in more specific agriculture-based works. Therefore, our aim is to show and evaluate our approach, including the challenges in building datasets for data-driven machine learning.

Our contributions in this work are therefore threefold:

- Present our annotation process for WPCS with respect to kernel fragmentation and stover overlengths.
- Show an analysis of the quality and consistency of the resulting annotations.
- Evaluate SSL for WPCS showing a considerably more efficient alternative to manual annotations for supervised learning.

## 2 Related Work

To the best of our knowledge there does not exist any image datasets for WPCS. Therefore, we investigate dataset creation in regards to benchmark datasets for both agriculture and in general object recognition. Starting with the latter there exists a large number of public datasets in the computer vision domain. For example, *paperswithcode*<sup>1</sup> lists 160, 195, and 37 for object detection, semantic segmentation, and instance segmentation, respectively. Larger benchmark datasets have the ability to form golden standards in the computer vision community and can be used to evaluate algorithms and push overall research.

Common among them is the aim to have a dataset with high quality and consistent annotations often over hundreds of thousands of images and hundreds of potential classes. The process for creating such datasets is expensive

---

<sup>1</sup><https://paperswithcode.com/datasets>

and therefore requires an efficient and clear pipeline. Typically a team of workers, either internal or outsourced, are instructed to annotate following a multiple stage pipeline aimed to maximise consistency and coverage. For example, in the ImageNet [4] object detection challenge a multi-stage solution first determined which object classes were present in a given image using a query-based algorithm to quickly traverse the 200 potential classes. Given these image-level class definitions an annotator is given a batch of images and instructed to draw a single bounding-box before moving to the next image. An image continues in this process until all bounding-boxes are annotated. Bounding-box quality and coverage are iteratively checked by another worker and once both pass the image is accepted into the dataset. Another multi-stage example is for the instance segmentation annotations in COCO [5] where images are annotated in three steps. First, an annotator determines if an object instance is present from a number of pre-defined super-categories, if yes, a symbol for a each specific sub-category is dragged and placed on a single instance. Next, each instance of every sub-category is marked until all object instances are covered. Finally, instance segmentation masks are annotated for each of the marked instances. During the final stage an annotator is asked to only annotate a single mask. Additionally, they are informed to verify previous segmentation annotations from other workers. In LVIS [11] the creators adopt a similar iterative pipeline to COCO of first spotting single object classes per image, followed by exhaustively marking each instance of a given category. In the next stage instance markings are upgraded to segmentation masks before moving on to verification. In a final stage, negative labels are added to the image. A three stage approach is used in Objects365 [3] where first non-suitable iconic images are filtered, iconic images typically only have a single clear object in the middle of the image and are deemed to be too simple. Next image-level tags are added based on super-categories, followed by the final step of annotating all bounding-boxes into sub-categories. There are also examples of creating datasets through less-defined processes but rather attempt to conduct the annotations closer to the expert knowledge, however, this is less common as the size of datasets are becoming increasingly larger. For example, in PASCAL VOC [12] the initial annotations were done by researchers at a single annotation event. While in ADE20K [13] the dataset is ambitiously annotated by a single person aiming to maximise consistency.

Common to most benchmark datasets is the usage of various roles that often require training. The role of an annotator is naturally used in all benchmarks and a training task is given to evaluate their ability. For example, in ImageNet [4] annotators must pass a drawing and quality verification test. In both tests the aim is to learn three core rules, for example, for drawing boxes only the visible parts should be annotated as tightly as possible. Multiple roles can add further verification such as in Objects365 [3]. Here, a course



## 2. Related Work

must be taken to learn how to become an annotator or inspector. Annotators are trained to draw bounding-boxes and inspectors to verify all annotated images. Furthermore, an examiner role is also included to review output from the inspector.

Finally, the usage of a golden standard set, where annotations are verified by experts to be near 100% accurate are used throughout almost all of the benchmarks. In LVIS [11] gold sets are added in multiple places in the pipeline and further work is prioritised to reliable workers. In ImageNet [4] they are used as overall quality control and also during training of annotator and inspectors.

While procedures such as multi-stage pipelines, training and roles can be important there also exists alternative approaches to either aid annotators or speed-up the tasks. This can be especially useful if dataset creators do not have a large amount of resources to implement the points covered so far. Researchers have investigated how to make the process of drawing annotations on an image more efficient. For example, Extreme Clicking was introduced in [14] and used to annotate the Open Images dataset [15]. Extreme Clicking allows for fast drawing by having the user click the four most extreme points of an object. It was found to decrease the drawing time from 25.5 seconds to 7.4 seconds in comparison to traditional box dragging. Annotation tools can also be enhanced by allowing the tool to produce annotations which a user can adjust [16, 17]. Another alternative is weak supervision that takes lower quality labels and is able to transfer this knowledge into the training. Such approaches use features from models pre-trained on larger datasets to train a classifier from or models can be finetuned towards a more specific task [6]. A popular approach in SSL is to increase the amount of labelled data by using pseudo labels from a fully-supervised model where a teacher network trains a student network. This approach has been popular in classification tasks but less so in object detection and segmentation, as the latter tasks are often more challenging due to the often large class imbalance between background and foreground objects [18]. However, recent works exist that aim to take these and additional challenges into account [18, 19].

Within agriculture there exists a number of datasets for different applications. These are extensively covered in [20] and we use this work as inspiration to analyse the dataset creation in similar applications to our work. For most agriculture dataset papers there is minimal description of the process of conducting annotation. Most simply state that object instances were annotated in either a bounding-box or mask format. In some cases there is a description of an open-source annotation tool but without stating details, e.g. the MangoNet dataset [21]. However, a few provide details on the specific tool, including DeepSeedling [22], where a dataset of bounding-boxes for cotton seedlings is collected using MS VoTT. Also in the MineApple dataset [23] the VIA annotation tool is used to annotate apples with bounding boxes. Fi-

nally, in DeepFruits [24] a custom MATLAB annotation tool is produced and has been publicly released by the authors.

As mentioned, the process for collecting and conducting annotation is rarely covered apart from a couple of datasets. An exception is the MineApple dataset [23], here an annotation worker is first instructed how to annotate before they can perform the task and after annotating an initial ten images are given in-person feedback. Furthermore, verification of all annotations is done to correct annotations from the workers. The process is also briefly described for annotating corn tassels in [25] where annotators are given a training page before starting and gold standard sets are used to evaluate resulting annotations.

Lastly, a number of the datasets adopt tools that counteract manual annotation. In the Orchid fruit dataset [26] a custom tool is able to train and test in parallel during annotation, allowing to easily determine changes in accuracy as additional examples are added to the dataset. In the Fruit Flowers dataset [27] the annotation tool FreeLabel [28] aided by having the worker draw freehand on a tablet for regions that contained flowers and the tool generated masks using region growing refinement. Finally, synthetic annotation have been used for the GrassClover dataset [29], by pasting plant crops onto background images of soil while randomly sampling rotation and scale in addition to adding shadows to the crops.

### 3 Dataset Annotation

In this section we present an overview of our process for creating annotated datasets for WPCS. Two different forms of dataset are created, one for kernel fragmentation and another for stover overlenghts. For each we cover our annotation guidelines for annotators, present statistics over datasets, and present an evaluation of the quality and consistency of annotations.

#### 3.1 Kernel Fragmentation

As mentioned, the datasets for kernel fragmentation have been previously used in a number of works [1, 2, 10]. The works showed for a number of deep learning models the potential of measuring kernel fragmentation in non-separated samples. In [2] the trained models were additionally evaluated against physically sieved samples for CSPS, showing a strong correlation. However, the best performing models between annotation-based metrics and CSPS correlation were not always consistent. Therefore, in this section we present and evaluate our process for annotating kernel fragmentation in our images.

### Annotation Guideline

To solve the task of estimating kernel fragment quality, the aim was to annotate all fragments allowing for an estimation of an industry standard such as CSPS. This would ideally allow a system to learn and estimate from images the differences in fragmentation given the condition present to a farmer's field. Figure E.2 shows fragment annotations in two cases with a clear difference in fragmentation. Both images are captured in the same field and have an identical TLOC but with different PGs. A PG of 1 mm in E.2(a) produces a larger number of smaller fragments and fragments in total compared to E.2(b) harvested with PG 4 mm. It is worth stating that there is not necessarily such a significant difference in fragmentation, however, the general expectation is a larger number of fragments with a smaller size as the PG decreases.



**Fig. E.2:** The difference in kernel fragmentation potentially present in images between different PGs. Both samples are harvested with TLOC of 11.5 mm but (a) had a PG of 1 mm and (b) 4 mm.

In addition to informing annotators to annotate all fragments, a number of specific cases were also addressed that occurred due to working with non-separated samples. Firstly, despite working with a resolution of 20 pixels to 1 mm, very small fragments in images were both difficult to annotate and to determine if they were truly kernel fragments. Therefore, an indicator was added to the annotation tool with a radius of 1 mm showing the minimum size fragments should be before they are annotated. The indicator is shown in Figure E.3 together with a zoomed in view. The indicator followed the user's mouse cursor and if a fragment's axis extended beyond the diameter the user should start the annotation process for the instance.

Another specific case is when fragments are grouped closely, here it could be ambiguous whether these were a single fragment or where the boundary between them should be. Therefore, a number of examples, such as Figure E.4, was provided to annotators with the aim of providing guidance.

Finally, as we are working with non-separated samples, kernel fragments can be partially covered by other fragments or stover. Naturally, this is not ideal as the image is not able to provide a true description of the fragmenta-

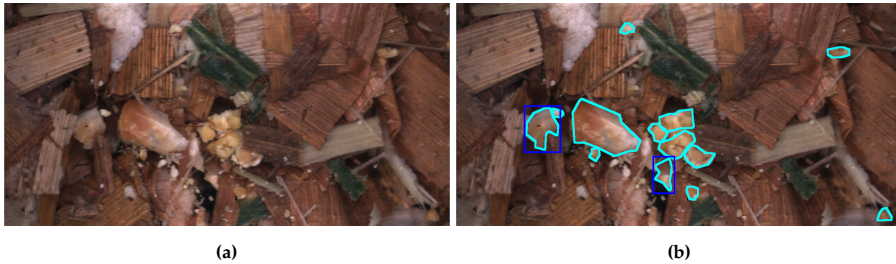


**Fig. E.3:** A blue indicator is shown indicating the minimum size of particles to be annotated.



**Fig. E.4:** An example of how to annotate fragments that are grouped closely together.

tion level for these cases. A solution could be for an annotator to estimate the true boundary, however, we determined this be difficult and potentially lead to errors when training the data-driven models. Therefore, annotators were instructed to only annotate the visible boundary. This is visualised in Figure E.5 with the original image in E.5(a) and two cases of annotations of covered fragments in E.5(b).



**Fig. E.5:** An example of how to annotate instances that are covered by other particles.

## Statistics and Evaluation

The annotation process was conducted over a number of iterations as images were gathered over harvest seasons. Therefore, we have split the data into a number of datasets that are named based on the harvest year. These could be used either individually or combined for a larger dataset during

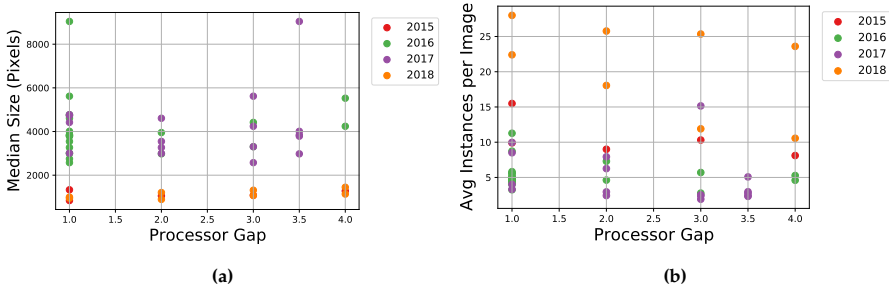
### 3. Dataset Annotation

model development. An overview of the annotation statistics for each harvest season can be seen in Table E.1, showing the machine setting the silage was harvested with (PG and TLOC), total number of images annotated, total instances annotated and average instances per image. The statistics are summarised for each PG, as this machine setting has the largest effect on fragmentation within a dataset. Additionally, if there are multiple harvest sequences of the same PG, these statistics are summarised in a single row where the number in the parenthesis shows the total number of sequences. Firstly, we can see that 2017 dataset has a significantly larger number of total images and instances compared to the three other datasets. While the annotation process was completed over a number of years, a comprehensive effort was made after this harvest to build a large dataset resulting in a skew towards this harvest. Secondly, the average number of annotated instances per image varies across the datasets, for example, between 2 to 8 instances in 2016 and 2017. Furthermore, a significant increase is seen in 2015 with 8 to 15 instances and in 2018 with 10 to 28 instances.

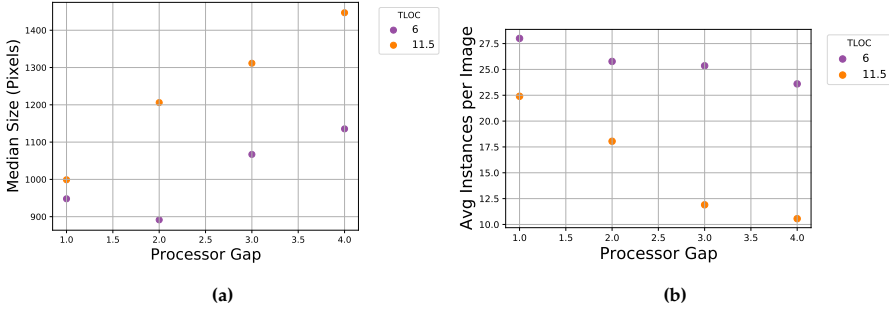
**Table E.1:** Annotation statistics for the images captured over four different harvest seasons.

PG	TLOC	Images	Anno Insts	Insts per Img
2015				
1 (2)	9	90	1333	14.8
2 (1)	9	21	189	9.0
3 (1)	9	37	402	10.31
4 (1)	9	39	300	8.11
Total		187	2224	11.89
2016				
1 (14)	4	131	762	5.82
2 (2)	4	18	110	6.11
3 (2)	4	19	82	4.32
4 (1)	4	11	58	5.27
Total		205	1118	5.45
2017				
1 (2)	4	152	967	6.36
2 (2)	4	127	458	3.61
3 (2)	4	359	901	2.51
3.5 (2)	4	126	442	3.51
1 (2)	12	290	1200	4.14
2 (2)	12	289	1909	6.61
3 (2)	12	111	927	8.35
3.5 (2)	12	171	435	2.54
Total		1972	8270	4.19
2018				
1 (1)	6	20	616	28.00
2 (1)	6	20	567	25.77
3 (1)	6	20	507	25.35
4 (1)	6	20	472	23.60
1 (1)	11.5	20	448	22.40
2 (1)	11.5	20	361	18.05
3 (1)	11.5	20	238	11.90
4 (1)	11.5	20	264	10.56
Total		169	3473	20.55

The differences in annotations are highlighted in Figure E.6 with the average size of annotated instances (a) and average number of instances per image (b) for each sequence. The expectation, at least within a harvest year, is that in general a smaller PG should produce smaller and more fragments compared to larger PG. For the datasets from 2015, 2016 and 2017 this trend is not overly clear in Figure E.6. However, the annotations from 2018 were done as a direct attempt to address this through a sanity check with a high requirement on annotation quality from a single annotator. This resulted in both a considerable increase in the average number of instances per image, as seen in Table E.1 and a clearer trend over PGs in corresponding Figures E.7(a) and E.7(b). Additionally, in these figures it can be seen the effect of the TLOC, where a shorter length affects fragments with smaller size and increase in instances.



**Fig. E.6:** Median size of annotations for sequences across PGs (a). Average number of annotated instance for sequences across PGs (b).



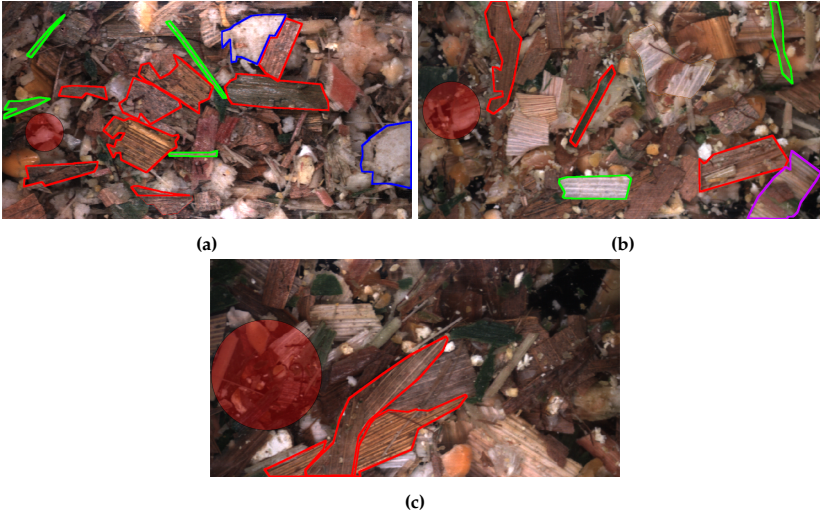
**Fig. E.7:** Statistics for annotations from 2018. Median size of annotations for sequences across PGs (a). Average number of annotated instance for sequences across PGs (b).

### 3.2 Stover Overlengths

In this section we cover the annotation process and statistics for determining stover quality. As covered in [2], we diverge from the kernel fragmentation strategy presented in the previous section and rather only aim to localise stover deemed as overlengths. An overlength per our definition is when a particle is  $1.5 \times \text{TLOC}$  or larger [2].

#### Annotation Guideline

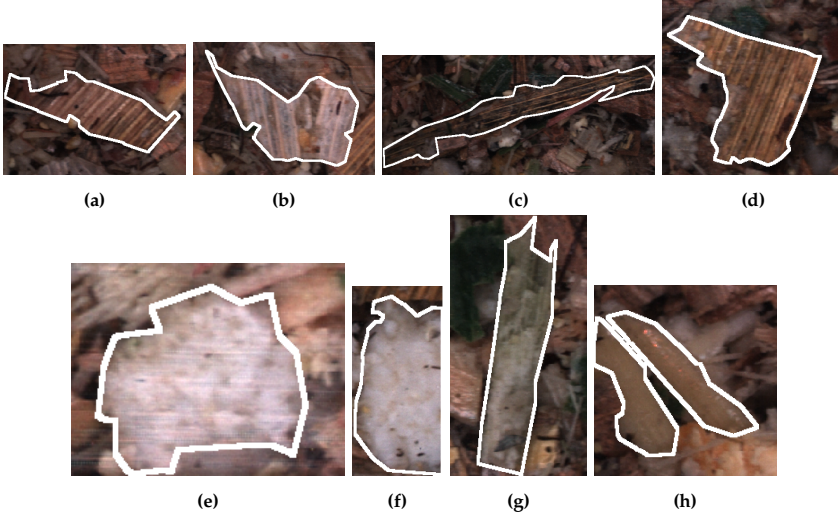
The differing overlength definition was visualised to the annotators through a red circle indicator as seen in Figure E.8. The indicator could be used to see if an instance should be annotated based on if it exceeded beyond the radius along any axis. The size of the red indicator is  $1.5 \times \text{TLOC}$  for a given image.



**Fig. E.8:** Differences in image content and annotations for three TLOC. In (a) samples are harvested with 4 mm, (b) with 6 mm and (c) with 11.5 mm. For each image the overlength definition of  $1.5 \times \text{TLOC}$  is shown by diameter of the red circle.

In addition to informing how to annotate an overlength particle the annotators were given similar instructions as those to kernel fragments. These include only annotating the visible portion of instances and annotating individual instances when multiple are tightly grouped. Finally, the annotators were given a number of example annotations aiming to cover both the inter- and intra-class variance. Figure E.9 shows two examples of each class from image sequences captured at TLOC 4 mm. In Figure E.9a-b the accepted leaves class is shown, which occurs when an instance is an overlength but only based on the axis length that is perpendicular to the leaf structure. In

Figures E.9c-d, the counterpart to the previous class non-accepted leaves is presented. In this case the axis which follows the leaf structure exceeds the overlength definition. Figures E.9e-f shows examples of inner stalks which often has a sponge-like texture. Lastly, Figures E.9g-h covers two examples of outer stalks where there can be some variance in the colour.



**Fig. E.9:** Overlength class examples, accepted leaves (a-b), non-accepted leaves (c-d), inner stalk (e-f), and outer stalk (g-h). Annotations examples are all from images captured of WPCS harvested at a TLOC of 4 mm.

## Statistics and Evaluation

The final annotations used in [2] were done by a single annotator and an overview of the annotation statistics can be seen in Table E.2. The table shows that in general there are more instances with a smaller TLOC, in addition to instances having a smaller size. Additionally, with the larger TLOC of 11.5 mm the annotations are limited for some classes, such as inner stalk.

**Table E.2:** Annotation statistics for the overlength dataset. Table also used in [2].

TLOC	Images	Instances	A leaves	NA leaves	Inner stalk	Outer stalk	Avg. size	Avg. major axis length	Avg. minor axis length
4	163	1233	520	419	75	209	14518.9	216.6	94.3
6	199	904	182	559	35	122	26315	294.3	122.7
11.5	113	263	51	172	1	38	61328.2	485.5	179.9

Before defining the dataset shown in Table E.2 and used in [2], an initial annotation iteration was done by three annotators on images harvested with a TLOC of 4 mm. As seen for kernels, we observe an inconsistency between



### 3. Dataset Annotation

the annotators on metrics such as the number of instances and average size which we show in Table E.3 and across the overlenth classes in Table E.4.

**Table E.3:** Annotation statistics for overlenth for three different annotators. Each numbered sequence contains images harvested with the same machine settings.

	Images	Instances	Avg. insts per image	Avg. size	Avg. major axis length	Avg. minor axis length
Annotator 1						
Seq1	37	73	1.97	33056.85	322.33	140.05
Seq2	32	57	1.78	36415.78	360.33	124.02
Seq3	31	102	3.29	25180.66	292.98	124.02
Annotator 2						
Seq1	37	124	3.35	25423.53	294.71	126.33
Seq2	32	180	5.62	20969.54	262.65	111.25
Annotator 3						
Seq1	37	271	7.32	17105.99	234.34	102.44
Seq2	32	256	8.0	18098.88	232.60	111.25
Seq3a	31	227	7.32	18025.16	234.55	111.41
Seq3b	31	222	7.16	18427.43	242.06	110.28

**Table E.4:** Class instances annotated by the three annotators for stover overlenth.

Annotator	A Leaves	NA Leaves	I Stalks	O Stalks
Annotator 1	122	46	7	7
Annotator 2	82	98	10	24
Annotator 3	418	330	65	157

We also had the annotators annotate some overlapping images over the three sequences. In Seq1 and Seq2 a total of 10 and 5 images were annotated respectively by all three persons. Whereas, in Seq3 5 images were annotated by both annotator 1 and 2. An analysis of the inter-rater agreement using Cohen’s Kappa coefficient [30] confirms that there is little agreement as seen in Table E.5. Cohen’s Kappa is a statistic that can measure the reliability of two persons annotating the same instances while taking into account that the agreement could be by chance. We define an annotation to be an agreement when two polygon annotations have an Intersection-over-Union (IoU) greater than 0.5. Table E.5 shows that for each sequence pair, the agreement scores 0 which can be interpreted as no agreement. Additionally, in the right portion of the table we show for a given annotator the number of annotated instances and the number of agreed annotations per counterpart annotator.

Based upon the above observations we perform an additional experiment to highlight the potential pitfalls of using inconsistent annotations by training two different models. Firstly, we focus on how well the model performed in terms of precision and recall. But also on what effect this has when evaluating

**Table E.5:** Cohen Kappa Score between each annotator pair with a single annotator as reference annotator (left most column). Additionally, the total number of instances per reference annotator with counts of overlap where IoU is greater than 0.5 for each sequence.

Cohen Kappa			Count IoU >0.5			
A1	A2	A3	Inst Cnt	A1	A2	A3
Seq1	0	0	25		1	0
Seq2	0	0	6		0	0
Seq3	0	na	15		na	15
A2	A1	A3	Inst Cnt	A2	A1	A3
Seq1	0	0	62		7	4
Seq2	0	0	23		3	1
Seq3	na	na	na		na	na
A3	A1	A2	Inst Cnt	A3	A1	A2
Seq1	0	0	78		9	6
Seq2	0	0	37		4	3
Seq3	0	na	39		3	na

with a test set if annotations are not consistent in training. It can be challenging to optimise a model when annotations are not consistent. However, it is also difficult to determine if alterations to a model improve or worsen if the basis of false positives and true positives are incorrect during testing. Therefore, models were trained on the two datasets of different consistency, namely a Faster R-CNN [31] with an Inceptionv2 [32] backbone using transfer learning from COCO using the TensorFlow Object Detection API [33]. This is the same training strategy used for baseline overlength models in [2].

In Table E.6 we show Average Precision (AP) and Average Recall (AR) results based on COCO standards [5] on a test set with inconsistent annotations. For each metric we show two values, the upper is a model trained on inconsistent annotations from all three annotators and the lower trained on consistent annotations from Annotator 3. In both cases the annotations are split 70% for training, 15% for validation and 15% for testing. Additionally, the splits from Annotator 3 were the same across both datasets to ensure comparable results. Table E.6 shows when looking at all classes that the model trained on consistent data performs in general a number of percentage points (p.p.) higher but scores lower on AR when more predictions are allowed. There is a clear difference between the two models when evaluating inner stalk predictions, here, the model trained on consistent data scores between 20 to 30 p.p. higher.

Clearer results can be seen when evaluating on the consistent test set in Table E.7. Increases in AP can be seen for the consistent-trained model, with AP@0.75 rising by almost 15 p.p.. For individual classes significant increases

### 3. Dataset Annotation

**Table E.6:** Results on test set with inconsistent annotations from the three annotators. For each metric results from two models are shown trained on different sets of data, upper is that on the inconsistent and below is trained on consistent.

Class	AP	AP@0.5	AP@0.75	AR@1	AR@10	AR@100
All (207)	23.7	42.2	25.8	23.0	<b>42.5</b>	<b>47.8</b>
	<b>28.1</b>	<b>48.1</b>	<b>34.8</b>	<b>26.9</b>	42.1	45.5
A Leaves (107)	29.1	<b>47.3</b>	33.6	17.4	51.6	57.8
	<b>29.2</b>	41.8	<b>39.6</b>	<b>17.7</b>	<b>55.7</b>	<b>61.3</b>
NA Leaves (59)	17.9	<b>34.2</b>	17.4	19.3	<b>35.3</b>	<b>44.4</b>
	<b>20.0</b>	<b>34.2</b>	<b>21.2</b>	<b>22.8</b>	30.6	35.6
I Stalks (11)	31.7	54.7	31.3	27.3	50.9	50.9
	<b>51.7</b>	<b>76.3</b>	<b>59.2</b>	<b>34.6</b>	<b>51.8</b>	<b>55.4</b>
O Stalks (30)	<b>15.9</b>	<b>32.7</b>	<b>20.8</b>	28.0	<b>32.3</b>	<b>38.0</b>
	10.0	30.6	19.0	<b>32.3</b>	23.3	29.7

are seen for all classes except outer stalks in terms of AP.

**Table E.7:** Results on test set with consistent annotations from the one annotator. For each metric results from two models are shown trained on different sets of data, upper is that on the inconsistent and below is trained on consistent.

Class	AP	AP@0.5	AP@0.75	AR@1	AR@10	AR@100
All (141)	32.0	54.2	35.6	23.8	45.1	<b>49.1</b>
	<b>39.7</b>	<b>63.0</b>	<b>50.4</b>	<b>29.6</b>	<b>46.0</b>	46.9
A Leaves (64)	44.6	70.7	54.7	20.2	55.8	61.1
	<b>49.1</b>	<b>70.8</b>	<b>68.5</b>	<b>22.6</b>	<b>60.8</b>	<b>63.9</b>
NA Leaves (43)	23.5	45.0	19.6	17.2	<b>35.6</b>	<b>43.5</b>
	<b>27.8</b>	<b>48.7</b>	<b>25.5</b>	<b>23.7</b>	29.3	34.4
I Stalks (10)	37.4	64.5	36.9	30.0	56.0	56.0
	<b>60.5</b>	<b>97.5</b>	<b>69.0</b>	<b>38.0</b>	<b>68.0</b>	<b>61.0</b>
O Stalks (24)	<b>22.5</b>	<b>36.6</b>	31.3	27.9	<b>32.9</b>	<b>35.8</b>
	21.1	35.1	<b>38.5</b>	<b>34.1</b>	25.8	28.3

Tables E.6 and E.7 show the importance of having consistent data when training models but also evaluating them. In both tables it can be seen that in general the model trained on consistent annotations have a higher AP compared to the inconsistent counterpart. Also, for the inconsistent model in Table E.7 the AP metrics are increased significantly in comparison to Table E.6. Therefore, if the model was evaluated on inconsistent annotations a conclusion could be made that the model performs poorly.

## 4 Semi-Supervised Learning

Due to the challenges and inconsistencies between annotators we perform investigations into the potential of using SSL to complement manual annotation for our dataset. We adopt the Unbiased Teacher methodology [18] due to their recent improvements with SSL for object detection. SSL has not been as extensively used in object detection tasks in comparison to classification, as there is often being a significant bias towards background in comparison to foreground. Therefore, the usage of pseudo labelling between a teacher and student network can be prone to learning a bias towards easier objects. However, with the Unbiased Teacher [18] the authors identify that in two-stage recognition networks, such as Faster R-CNN, the overfitting occurs in the classification heads for both the Region Proposal Network and final multi-class classification. The approach proposes to train a student and teacher mutually where the student learns from the teacher via highly augmented images and the teacher learns slowly from the student with an Exponential Moving Average (EMA). In addition to EMA, the framework adopts focal loss to concentrate on more challenging examples in order to lower the bias towards easier examples. The framework has a number of parameters that must be tuned in order to allow the two networks to improve together. Firstly, a confidence threshold that defines which predictions from the teacher are passed as annotated examples to the student. Second, the number of unsupervised images per iteration to create pseudo labels from. A variable controlling how much weight the unsupervised examples have when calculating loss. Finally, a number of burn-in iterations must be set where the teacher network is trained in order to provide a solid baseline before performing SSL.

We evaluate the usage of SSL by training teacher-student networks on two different annotated datasets together with a large number of unannotated images for kernel fragmentation. This includes the 151617 dataset presented earlier and used in a number of previous works [1, 2, 10] and a subset only including annotations from 2016. The 151617 training set includes 1393 images containing 6907 instances and the 2016 subset has 115 images with 675 instances. Our unsupervised portion of the SSL dataset are 7888 images from a harvest captured in 2019. Finally, we evaluate our SSL-trained models with both object detection metrics and correlation analysis against physically sieved CPCS samples first presented in [2]. For a stronger evaluation we use a new test set compared to previous WPCS works, adopting the sanity checked annotations from 2018 presented in Table E.1 and Figure E.7. This way we allow for less precise annotations during the training process but test networks against annotations of higher quality.

Our teacher-student networks follow the investigations done in [18] which are a Faster R-CNN [31] with an ResNet50 [34] Feature Pyramid Network [35]

backbone. Networks are trained on an NVIDIA Titan XP GPU using the Detectron2 framework [36]. The networks are trained for a total of 50000 iterations with a learning rate of 0.01 using Stochastic Gradient Descent with an initial burn-in of 10000 iterations. We use an EMA of 0.9996, following the value from the original authors which we also empirically determined to lead to stable results. A lower EMA would allow the student to contribute more during updating of the teacher network and may cause worse performance due to too noisy labels [18]. Finally, we also train baseline Faster R-CNN models in a fully supervised manner to compare our SSL models against.

In Table E.8 the results can be seen for a number of different teacher-student variants, additionally, the baseline model can be seen in the first row where the parameters for SSL are not applicable. The remaining rows show SSL training runs with all combinations of the three key SSL parameters. The confidence threshold for pseudo labels is set between 0.1 to 0.7 with 0.2 increments. The number of unsupervised images per iteration and how much weight to place on the unsupervised loss is set at either 1 or 4. For each SSL-trained model we evaluate the AP and Pearson’s Correlation Coefficient (PCC) with the network iteration with the lowest validation loss. Also shown in the table is that two SSL training runs diverged early and therefore results are not shown. In regards to AP metrics we see that SSL models trained with a bounding-box confidence threshold of either 0.3 or 0.5 improve results in comparison to the baseline model. The best performing model for AP and AP@0.5 are seen with a confidence threshold of 0.5, using 4 unsupervised images and an unsupervised weight of 4. Concretely, the AP is improved by 3.55 p.p. and AP@0.5 by 6.2 p.p.. At the more stringent AP@0.75 the network trained with the same parameters apart from a confidence threshold of 0.3, has a slight improvement with 4.51 p.p.. The PCC analysis can be seen in the three right-most columns in Table E.8 and we see that the best performing models for AP does not translate to improvements in PCC. However, the PCC is improved for CW43 by 0.04 and when combining the two harvest weeks by 0.07.

In Table E.8 we applied SSL in combination to the 151617 dataset which required a relatively large amount of effort in obtaining the initial 6907 annotated object instances. Therefore, in Table E.9 we investigate whether much less effort can be used and therefore only use the annotations from 2016 containing 675 instances. The unsupervised portion is extended to also include the images from 2015 and 2017 from the 151617 dataset. This means that 1.4% of the dataset in Table E.9 is annotated compared to 15.1% in Table E.8. The baseline model shows a considerable drop in AP and PCC in comparison to previous results. For example, AP decreases by 12.23 p.p. and 15.78 p.p. to 4.97 in comparison to the baseline and best performing model using 151617 as supervised labels. However, the teacher-student training with additional unsupervised data improves the baseline by a large margin.

**Table E.8:** Results for SSL-trained models for models with various hyper-parameters on the 151617 annotated dataset together with unannotated images from a harvest from 2019. Additional results also shown for baseline fully supervised models in the first rows.

Train Set	Unsup. Set	Bbox Thresh	Unsup Images	Unsup Weight	AP	AP@0.5	AP@0.75	PCC CW40	PCC CW43	PCC CW40+43
151617 [2]	NA	NA	NA	NA	NA	NA	NA	<b>0.95</b>	<b>0.79</b>	<b>0.81</b>
151617	NA	NA	NA	NA	17.20	32.15	15.96	0.94	0.75	0.68
151617	2019	0.1	1	0.5	15.57	27.73	16.16	0.88	0.73	0.72
151617	2019	0.1	1	4	-	-	-	-	-	-
151617	2019	0.1	4	0.5	17.49	31.36	17.40	0.86	0.76	0.71
151617	2019	0.1	4	4	-	-	-	-	-	-
151617	2019	0.3	1	0.5	17.85	31.92	17.78	0.93	0.72	0.75
151617	2019	0.3	1	4	19.93	36.02	19.64	0.81	0.74	0.67
151617	2019	0.3	4	0.5	17.95	32.32	17.66	0.92	0.71	0.72
151617	2019	0.3	4	4	19.73	35.15	<b>20.47</b>	0.85	0.69	0.65
151617	2019	0.5	1	0.5	19.79	35.86	20.32	0.90	<b>0.79</b>	0.70
151617	2019	0.5	1	4	17.78	34.85	15.45	0.83	0.73	0.62
151617	2019	0.5	4	0.5	19.66	36.45	18.54	0.88	0.72	0.65
151617	2019	0.5	4	4	<b>20.75</b>	<b>38.35</b>	19.99	0.88	0.72	0.63
151617	2019	0.7	1	0.5	15.56	27.82	15.36	0.88	0.63	0.63
151617	2019	0.7	1	4	15.36	28.13	15.13	0.77	0.58	0.59
151617	2019	0.7	4	0.5	15.47	28.48	14.84	0.86	0.60	0.58
151617	2019	0.7	4	4	13.58	24.37	13.22	0.77	0.55	0.56

The SSL model trained with 0.7 confidence threshold, 4 unsupervised images and an unsupervised weight of 4, increases AP by 12.69 p.p, AP@0.5 by 26.52 p.p. and AP@0.75 by 10.19 p.p. The same model increases the PCC for both harvest weeks from 0.56 to 0.64. However, this improvement appears to be present largely for the first week as better PCC can be seen for another teacher-student training at 0.1 bounding-box threshold. Overall the AP and PCC results are not improved in comparison to those in Table E.8, however, significant effort in annotation could have been saved using this approach.

## 5 Discussion

Despite implementing annotations guidelines and using subject-matter experts as annotators we founds variation and inconsistencies. This shows both the difficult task of annotating our images and of manual annotation in general. The annotations could likely be improved with increased processes such as multiple annotation iterations per image/harvest and gold standard sets. However, these could be costly to implement and be time-consuming. We investigated a single alternative to manual annotation through a teacher-student training framework. Others could be of interest, such as an annotation tool aiding through automatic annotation.

Despite annotation inconsistency we still see a strong correlation in our models and in previous work. Therefore we suggest that the dataset is still suitable for training but care should be taken when evaluating models with

## 6. Conclusion

**Table E.9:** Results for SSL-trained models for models with various hyper-parameters on the 2016 annotated dataset together with unannotated images from harvests from 2015, 2017 and 2019. Additional results also shown for baseline fully supervised models in the first rows.

Train Set	Unsup. Set	Bbox Thresh	Unsup Images	Unsup Weight	AP	AP@0.5	AP@0.75	PCC CW40	PCC CW43	PCC CW40+43
2016	NA	NA	NA	NA	4.97	7.39	6.05	0.70	0.54	0.56
2016	1517+2019	0.1	1	0.5	11.95	24.02	9.65	0.72	0.65	0.63
2016	1517+2019	0.1	1	4	-	-	-	-	-	-
2016	1517+2019	0.1	4	0.5	14.24	28.51	11.51	0.70	<b>0.76</b>	0.59
2016	1517+2019	0.1	4	4	12.00	22.43	11.32	0.74	0.66	0.65
2016	1517+2019	0.3	1	0.5	13.67	27.15	10.89	0.70	0.57	0.52
2016	1517+2019	0.3	1	4	-	-	-	-	-	-
2016	1517+2019	0.3	4	0.5	13.28	27.90	9.35	0.85	0.55	0.62
2016	1517+2019	0.3	4	4	13.53	24.15	13.31	0.84	0.66	0.70
2016	1517+2019	0.5	1	0.5	15.05	29.60	12.30	0.73	0.64	0.56
2016	1517+2019	0.5	1	4	-	-	-	-	-	-
2016	1517+2019	0.5	4	0.5	16.98	33.60	13.98	0.83	0.70	<b>0.71</b>
2016	1517+2019	0.5	4	4	16.62	32.75	14.16	0.79	0.65	0.58
2016	1517+2019	0.7	1	0.5	12.34	21.14	13.52	0.82	0.55	0.64
2016	1517+2019	0.7	1	4	13.92	27.88	11.91	0.85	0.61	0.58
2016	1517+2019	0.7	4	0.5	9.67	16.23	10.59	0.74	0.59	0.64
2016	1517+2019	0.7	4	4	<b>17.66</b>	<b>33.91</b>	<b>16.24</b>	<b>0.90</b>	0.61	0.64

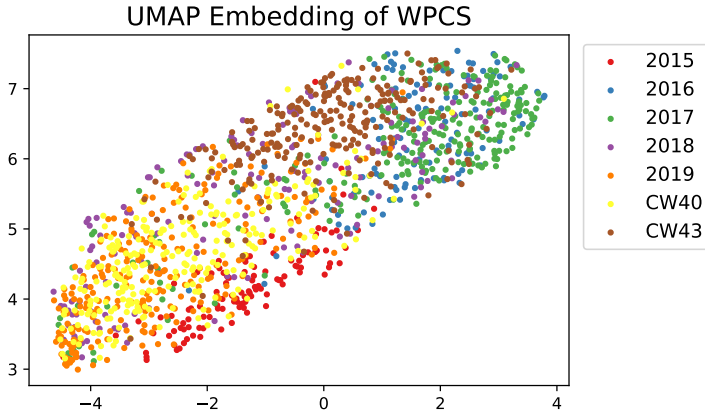
annotation-based metrics such as AP. Instead, this should be done in conjunction with physically-sieved estimates such as CSPS.

In our datasets, especially for kernel fragments, bias has been attempted to be counteracted by including images from multiple different harvest seasons and machine settings. There is likely considerable variation in WPCS and the resulting images. If a system were to be evaluated across thousands of farms all over the world, care should be taken for additional datasets to take this into account. For examples, in Figures E.10 we show the UMAP [37] embeddings of a random sample of up to 250 images from our images over the multiple harvests. We see that the RGB embeddings do cluster and this information could be utilised.

## 6 Conclusion

The majority of deep learning methods are reliant on annotation. This can be difficult and expensive for more specific applications, such as within agriculture. Annotation process is often not covered in such datasets making it difficult to reproduce or evaluate the research fully. Therefore, the aim of this work was to describe a concrete case and thereby illustrate the actual challenges and how we have addressed them.

In this work we have presented for WPCS our annotation process, statistics and an analysis of our datasets which is not often done in specific use-cases within agriculture. Manual annotation is often a challenging and time-consuming task which has been the case in our dataset seen by variations



**Fig. E.10:** UMAP Embeddings of various RGB images captured during different harvests.

in statistics for the annotations between annotators and between harvest seasons.

We evaluate the usage of SSL, with a teacher-student approach as an extension to manual annotations. Our SSL-trained object detectors showed promise by increasing AP but no significant alteration when evaluating CSPA against physical samples. However, we did see significant improvements when using the approach on a much smaller annotated set from a single harvest season.

We hypothesise that a combination of increased processes and further alternative tools can significantly decrease the annotation cost as larger datasets would be required to cover additional variations in farms. We believe that exploring challenges in smaller datasets is a crucial step in all domains. Being aware or addressing them to improve the overall quality is crucial for success, whether it be training successful models or having the ability to evaluate them with annotation-based metrics.



## References

- [1] C. B. Rasmussen and T. B. Moeslund, "Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images," *Sensors*, vol. 19, p. 3506, 08 2019.
- [2] C. B. Rasmussen, K. Kirk, and T. B. Moeslund, "Anchor tuning in faster r-cnn for measuring corn silage physical characteristics," *Computers and Electronics in Agriculture*, vol. 188, p. 106344, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169921003616>
- [3] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8429–8438.
- [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 1–42, 01 2015.
- [5] 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 *Proceedings of the European conference on computer vision*, 2014, pp. 740–755.
- [6] D. Zhang, J. Han, G. Cheng, and M.-H. Yang, "Weakly supervised object localization and detection: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [7] B. H. Marsh, "A comparison of fuel usage and harvest capacity in self-propelled forage harvesters," *International Journal of Agricultural and Biosystems Engineering*, vol. 7, no. 7, pp. 649 – 654, 2013. [Online]. Available: <https://publications.waset.org/vol/79>
- [8] D. Mertens, "Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages," In: *Proceedings of the Four-State Dairy Nutrition and Management Conference*, 01 2005.
- [9] J. Heinrichs and M. J. Coleen, "Penn state particle separator," May 2016. [Online]. Available: <https://extension.psu.edu/penn-state-particle-separator>
- [10] C. B. Rasmussen and T. B. Moeslund, "Evaluation of model selection for kernel fragment recognition in corn silage," <https://arxiv.org/abs/2004.00292>, 2020.

## References

- [11] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [12] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," 2010.
- [13] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5122–5130.
- [14] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4940–4949.
- [15] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, pp. 1956–1981, 2020.
- [16] L. Castrejón, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *CVPR*, 2017.
- [17] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," 2018.
- [18] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, "Unbiased teacher for semi-supervised object detection," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [19] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, Y. J. Lee, A. G. Schwing, and J. Kautz, "Instance-aware, context-focused, and memory-efficient weakly supervised object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [20] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Computers and Electronics in Agriculture*, vol. 178, p. 105760, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920312709>
- [21] R. Kestur, A. Meduri, and O. Narasipura, "Mangonet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 59–69, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197618301970>

- [22] Y. Jiang, C. Li, A. H. Paterson, and J. S. Robertson, "Deepseedling: deep convolutional network and kalman filter for plant seedling detection and counting in the field," *Plant Methods*, vol. 15, no. 141, 2019.
- [23] N. Hani, P. Roy, and V. Isler, "Minneapple: A benchmark dataset for apple detection and segmentation," 2019.
- [24] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016. [Online]. Available: <http://www.mdpi.com/1424-8220/16/8/1222>
- [25] N. Zhou, Z. D. Siegel, S. Zarecor, N. Lee, D. A. Campbell, C. M. Andorf, D. Nettleton, C. J. Lawrence-Dill, B. Ganapathysubramanian, J. W. Kelly, and I. Friedberg, "Crowdsourcing image analysis for plant phenomics to generate ground truth data for machine learning," *PLoS Comput Biol.*, vol. 14, no. 7, 2018.
- [26] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3626–3633.
- [27] P. A. Dias, A. Tabb, and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3003–3010, 2018.
- [28] P. A. Dias, Z. Shen, A. Tabb, and H. Medeiros, "Freelabel: A publicly available annotation tool based on freehand traces," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 21–30.
- [29] S. Skovsen, M. Dyrmann, A. K. Mortensen, M. S. Laursen, R. Gislum, J. Eriksen, S. Farkhani, H. Karstoft, and R. N. Jorgensen, "The grass-clover image dataset for semantic and hierarchical species understanding in agriculture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [30] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. [Online]. Available: <https://doi.org/10.1177/001316446002000104>
- [31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>

## References

- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.
- [33] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3296–3297.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 06 2016, pp. 770–778.
- [35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [36] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [37] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

# Paper F

## Evaluation of Edge Platforms for Deep Learning in Computer Vision

Christoffer Bøgelund Rasmussen, Aske Rasch Lejbølle, Kamal  
Nasrallohi and Thomas B. Moeslund

The paper has been published in  
Volume 12664 of the *Lecture Notes in Computer Science*, pp. 523-537, 2021.

© 2021 Springer  
*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 identify 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. Each year, the top-1 error on the large ImageNet dataset, used within object classification, gradually decreased [1, 2]. As other computer vision (CV) tasks gained more interest, such as object detection and semantic segmentation, accuracies on benchmark datasets would increase each year [3, 4]. However, recently, focus has shifted towards more practical usage of DNNs. Today, lowering the network complexity while maintaining a high accuracy is largely prioritised. Novel architectures are developed that contain fewer parameters [5, 6], 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 operate on 16-bit floating point (FP16), or even 8-bit integers, with almost no loss in precision [7].

Following the trend within academia of developing DNNs, 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 Amazon's Amazon Web Services (AWS) [8], have emerged that can train and run models online. Furthermore, Internet of Things (IoT) have resulted in products that require smaller and cheaper computers, which can run trained models at the edge. As a result of this demand, companies like Intel and NVIDIA have brought onto the market edge platforms that deploy and run network inference at limited costs [9, 10]. 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, detection and segmentation.

In this work, we evaluate edge platforms on common CV tasks, including object classification, object detection and semantic segmentation. 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 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 is a measure to identify the cost effectiveness of a certain platform/model combination, using a specific batch size. Additionally, comparing retail price and FPS, we propose a framework which aid the optimal platform/model selection, depending task and budget/speed requirements. Finally, we compare the distribution of DNN operations across platforms and models to identify the parts of a DNN on each platform that result in a higher FPS costs. These investigations allow us also to evaluate different CV tasks over the edge platforms and conclude on the best platform for a given use-case in terms of value for money, budget and FPS.

Previous works have studied models of different complexities [11–13], however, these publications aim to provide analyses of the speed/accuracy trade-off between models. On the other hand, works have been published that evaluates and compares different edge platforms [14, 15], 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, resources and the given CV task.

## 2 Related Work

### 2.1 Object Classification

In the last couple of years, works have been published that compare classification models and their performances. Canziani, Culurciello and Paszke [12] analysed inference time, power consumption and system memory utilisation for models of different complexity, depending on the batch size. However, all tests were performed on a single NVIDIA Jetson TX1, and only complex models were considered. Bianco et al. [11] extended the work of [12] by in-



## 2. Related Work

cluding several additional CNNs, while also performing the evaluation on an NVIDIA Titan X GPU, but considered the same parameters. Meanwhile, Velasco-Montero et al. [16] evaluated models of different complexities, 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. [14] conducted an evaluation of several classification models, including those in [11], but also less complex models. Furthermore, they considered five different platforms, including an edge platform. Similarly to [11, 12], 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. 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

Huang et al. [13] performed a comparison of three popular object detectors by changing the feature extractor, to analyse the change in accuracy/speed/memory trade-off. Liu et al. [17] presented a more extensive survey of object detectors, where less complex detectors were also considered. 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

Few works have been published in benchmarking of semantic segmentation networks. Guo et al. [18] 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 previously 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 complexities across different platforms. Trindade et al. [19] evaluated two popular frameworks, Caffe [20] and TensorFlow [21] and compared performance, with respect to training time, between a GPU and NUMA CPU. A more extensive

evaluation of frameworks was presented by Zhang, Wang and Shi [22] who performed the evaluation on different platforms where inference time, memory footprint and energy consumption was evaluated. Blouw et al. [15] measured inference time and energy consumption across different platforms 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. [23] focused on low-power devices, by evaluating object classification models and frameworks with respect to inference time and power consumption.

To our knowledge, only a single previous publication compares different platforms across different tasks, which is the aim this work. Ignatov et al. [24] considered mobile platforms containing chips that are manufactured by major chipset companies. 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 selected 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 together with their performance on leading benchmark challenges. 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 [21] framework. An overview of the models for each task can be seen in Table F.1.

#### Classification

We adopt MobileNetV1 [25] as the small, ResNet50 [26] as a medium, and InceptionResNetV2 [27] as the larger more complex network. An overview

### 3. Platform Evaluation

Model	Year	GFLOPS*	Top-1 [%]
MobileNetV1 [25]	2017	1.15	70.9
ResNet50 [26]	2015	6.97	75.2
InceptionResNetV2 [27]	2017	26.36	80.4
mAP [%]			
SSD MobileNetV1 [25]	2015	2.49	21
SSD InceptionV2 [13]	2017	9.63	24
mIOU [%]			
DeepLabV3 MobileNetV2 [6]	2018	17.69	75.32
DeepLabV3 Xception65 [28]	2017	354	82.20

\* As measured in TensorFlow

**Table F.1:** Overview of models over the three tasks. Top-1 accuracy is based on the ImagenNet classification task [29]. mAP is based on the COCO detection task [30]. mIOU is based on the VOC 2012 segmentation task [31].

of the classification models described can be seen in the top portion in Table F.1.

#### Object Detection

For benchmarking object detection networks, we use the SSD [32] with the distinction between the complexity of the SSD networks being done by switching the feature extractor. The middle portion in Table F.1 summarises our choices for the two feature extractors with varying complexity, namely, MobileNetV1 and InceptionV2.

#### Semantic Segmentation

We adopt DeepLabV3 [28] for evaluating semantic segmentation networks. An overview of model backbone choices for evaluation of DeepLabV3 is shown in the bottom portion of Table F.1, which in this case is MobileNetV2 and Xception65.

### 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 F.2, covering the number of cores, clock frequency, memory, Thermal Design Power (TDP) and price. We include a CPU, the Intel i7-7700K, since a GPU solution, occasionally, may not be possible due to price or space restrictions. Further, we

include two low-power edge devices, the Intel NCS and NCS2, that can perform inference of DNN models. The NCS devices have the form of USB sticks and must be connected to a host machine for inference. Additionally, we include an NVIDIA Jetson TX2, which requires more power, compared to the NCS devices, but is more powerful. Finally, we include a reference, NVIDIA GTX 1080, to which we can compare our evaluation of edge platforms.

Platform	Cores	Clock Freq. (GHz)	Memory (GB)	TDP (W)	Price* (\$)
i7-7700K	4	4.2	64	91	350
Intel NCS	12**	0.6	0.5	1	69
Intel NCS2	16**	0.7	0.5	1	75
NVIDIA GTX 1080	2560***	1.6	8	180	580
NVIDIA Jetson TX2	256***	1.3	8	7.5	560

\* Price per 01/09/2020 [33]    \*\* SHAVE cores    \*\*\* CUDA cores

**Table F.2:** Overview of evaluated platforms, including the reference GTX 1080.

### 3.3 Evaluation Overview

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 [34], 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 to FP16, with minimal loss in accuracy. To run inference on the NCS and NCS2, 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. This is accomplished using the OpenVINO toolkit [35], developed by Intel. Similarly to TF-TRT, this is done by fusion of certain layers of the network, such as Convolution and BatchNormalisation 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 optimise and run evaluation. However, TensorFlow 1.8 is used in case of TX2 as this is compatible with TensorRT 4.0.1, which is required to optimise 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

## 4. Experimental Results

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  $\leftarrow$   $\frac{\text{time()} - \text{start\_time}}{\text{batch\_size}}$  ;  
  | i  $\leftarrow$  i + 1 ;  
  | total_time  $\leftarrow$  total_time + inference_time ;  
mean_inference_time  $\leftarrow$   $\frac{\text{total\_time}}{100}$  ;
```

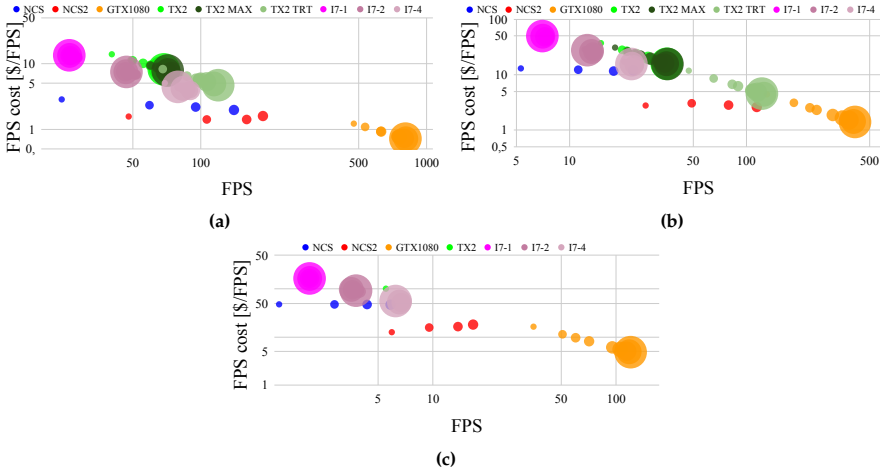
---

## 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 FPS cost in relation to the FPS of the difference platforms across the models of different complexities. The FPS cost is calculated as the retail price divided by the number of FPS for at given platform/model combination. Additionally, we plot price against FPS in order to show potential speeds based on concrete price points. Finally, we plot the top operation allocations for each platform to further understand the differences between the platforms. Since many of the plots show large numerical differences between platforms we plot values on a logarithmic scale.

## 4.1 Classification

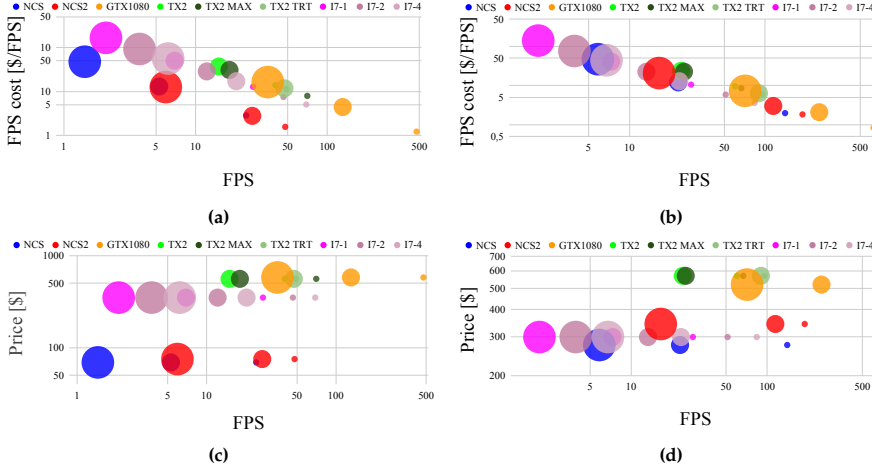
Figure F.1 shows the FPS cost for the three classification models over the platform variants. An increasing batch size is shown by an increasing diameter of the bubbles representing the platforms. It is clear that the NCS2 is the most cost friendly edge platforms between batch size 1 to 4. A difference in the NCS and NCS2 to the other platforms is the consistent costs over batch size as the number of sticks increases accordingly, whereas for the TX2 variants and GTX 1080 we see lower FPS cost as batch size increases. The i7 FPS cost does not change over batch sizes but does decrease as the number of cores is increased. The TX2 TRT does become competitive in comparison to the NCS2 for our medium complexity ResNet50 model at larger batch sizes.



**Fig. F.1:** FPS cost of classification models based on batch size and FPS. MobileNetV1 (a), ResNet50 (b) and InceptionResNetV2 (c).

Looking at the FPS for a given price point together with FPS cost in Figure F.2 we are able to determine the trade-off between model complexity, FPS and price budget for batch sizes 1 and 4. 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. Additionally, in these figures we depict the three models and their complexity by the size of the bubble. The lowest complexity MobileNetV1 is shown by the smallest diameter and most complex InceptionResNetV2 by the largest. For batch size 1 in Figure F.2 (a) and (c) the NCS and NCS2 are able to provide a relatively high FPS over the three model complexities for a low price point, furthermore, this is highlighted by the lower FPS cost. However, at batch size 4 the i7 CPU becomes more comparable in terms of price and, in the cases with less complex models and increased number of cores, have similar FPS to the NCS and NCS2.

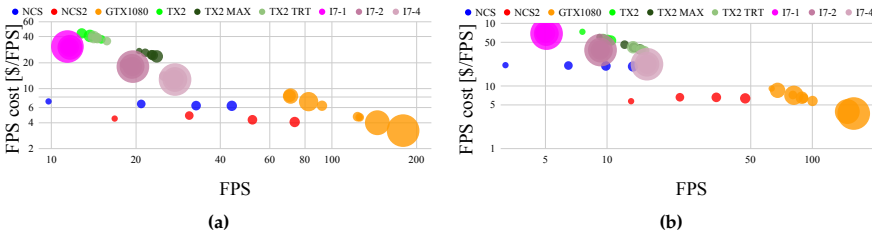
## 4. Experimental Results



**Fig. F.2:** Comparison of FPS cost (a) & (b) and retail price (c) & (d) based on FPS, for batch sizes one (a) & (c) and four (b) & (d). Small bubbles indicate MobileNetV1, middle-size bubbles indicate ResNet50, and large bubbles indicate InceptionResNetV2.

## 4.2 Object Detection

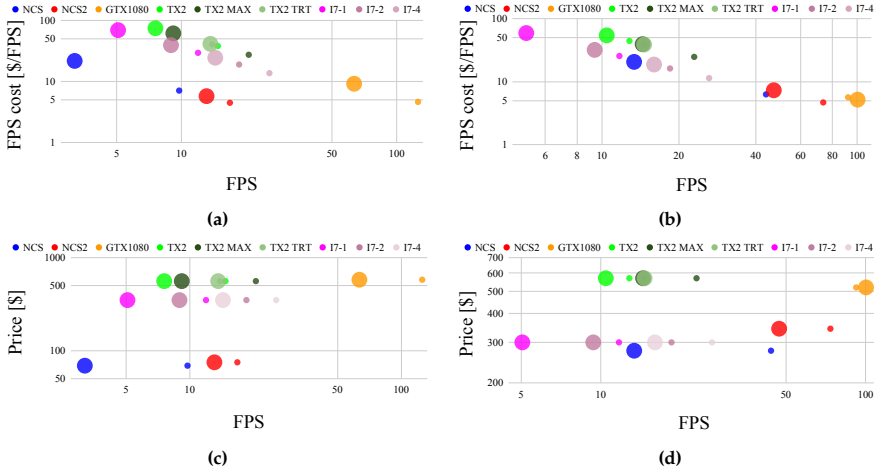
Again, we see in Figure F.3 that the NCS2 has the best FPS cost for the two SSD models, however, the NCS is competitive when MobileNetV1 is used as the backbone. In addition, despite the decreasing FPS cost as number of cores increase for the i7, or by increasing batch size and optimising for TX2 TRT, these variants are in general less viable for object detection purposes due to their overall FPS cost and lower FPS.



**Fig. F.3:** FPS cost of SSD MobileNetV1 (a) and SSD InceptionV2 (b) based on batch size and FPS.

Regarding the concrete price point and FPS cost, Figure F.4 shows that the NCS2 is the best trade-off at batch size 1 at a lower price and high FPS, additionally, the FPS cost is comparable to that of the GTX 1080. At batch size 4 the i7 is more competitive, especially as the number of cores increases but still has a lower FPS than the NCS2 where it is able to obtain impressive amounts of FPS at almost 50 FPS with InceptionV2 and around 80 FPS with

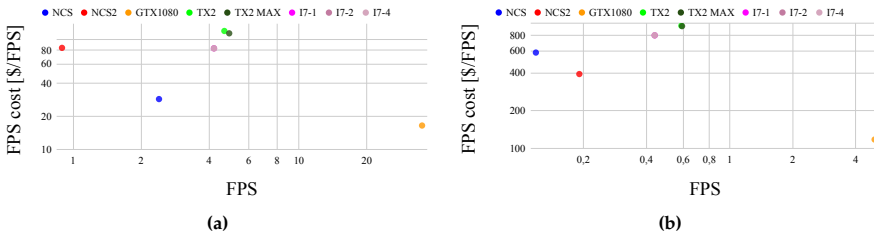
MobileNetV1. The FPS cost is again similar to that of the GTX 1080 for both SSD networks.



**Fig. F4:** Comparison of FPS cost (a) & (b) and retail price (c) & (d) based on FPS, for batch sizes one (a) & (c) and four (b) & (d). Small bubbles indicate SSD MobileNetV1 and middle-size bubbles indicate SSD InceptionV2.

### 4.3 Semantic Segmentation

It was only possible to run the DeepLabV3 models at batch size 1 due to memory constraints across the platforms. Figure F.5 shows that none of the edge platforms could run the models near real-time. For the DeepLabV3 with MobileNetV2 the NCS had a considerably lower FPS cost compared to the other platforms. Whereas, with Xception65 NCS2 was the best but still at a high FPS cost.



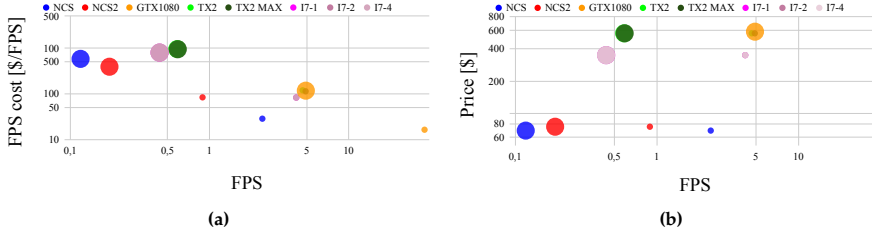
**Fig. F5:** FPS cost based on DeepLabV3 models based on batch size and FPS at batch size one. DeepLabV3 MobileNetV2 (a) and DeepLabV3 Xception65 (b).

Figure F.6 shows that the NCS and NCS2 have a low price point but also a low FPS. Only the TX2 and TX2 MAX for the MobilenetV2 variant show



## 4. Experimental Results

promise with almost 5 FPS but at price point similar to the of the GTX 1080. FPS cost depends on the complexity of the model, for the lower complex DeepLabV3 with MobileNetV2 the NCS is the clear cheapest, whereas, with Xception65 as the backbone FPS cost is largely similar but with NCS2 as the cheapest option.



**Fig. F.6:** Comparison of FPS cost (a) and retail price (b) based on FPS at batch size one. Small bubbles indicate DeepLabV3 MobileNetV2 and middle-size bubbles indicate DeepLabV3 Xception65.

### 4.4 Comparison of Tasks

We compare results from Figures F.2, F.4 and F.6 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

In order to understand more about the differences between the platforms we investigate the allocation of operations for the models. We use the TensorFlow profiler for the GTX 1080, TX2 and i7, whereas for the NCS and NCS2 we use the Deep Learning Workbench in OpenVINO. For each we visualise the operations as the top five and combine the remaining timings into one which we denote as *Other*. We only show the timings for the MobileNet variants from our three tasks in Figures F.7-F.9 as similar trends were seen for the other backbones. The top-5 operations are largely the same for the GTX 1080 and TX2. We see that the TX2 TRT bundles a significant number of operations in *TRTEngineOp* for the classification model but not so much for

the SSD variant. For all three tasks for the NCS and NCS2 a large portion is spent on the *Convolution* operation. Finally, the i7 is similar to that of the GTX 1080 and TX2 but does not show any type of convolution in the top-5.

### MobileNetV1 Operation Allocations

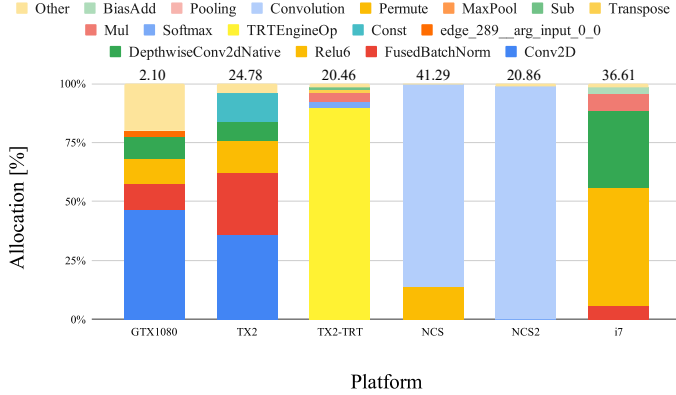


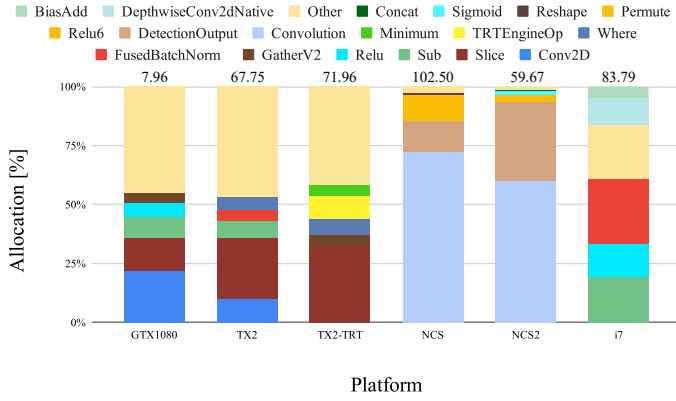
Fig. F.7: Operation allocation for MobileNetV1. Numbers above bars indicate total time in ms.

## 5 Conclusion

In this work, we have evaluated different edge platforms within object classification, object detection and semantic segmentation. We have analysed the FPS cost together with batch size and budget to aid decision making of platform/model selection. Finally, we have analysed allocation of DNN operations. 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

### SSD MobileNetV1 Operation Allocation



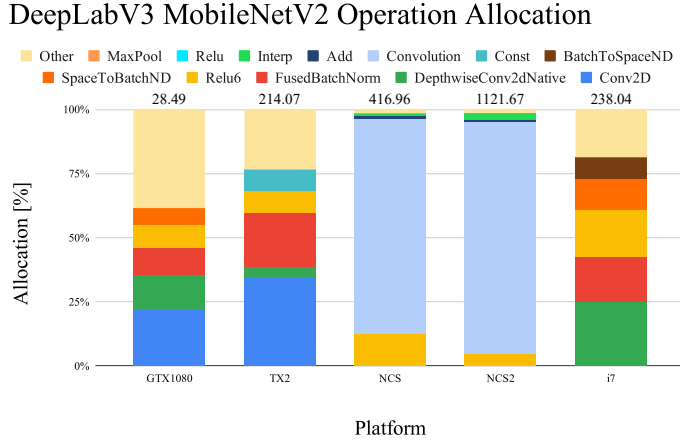
**Fig. F8:** Operation allocation for SSD MobileNetV1. Numbers above bars indicate total time in ms.

requirements.

Analysing 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 *IEEE conference on computer vision and pattern recognition*, 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.



**Fig. F9:** Operation allocation for DeepLabV3 MobileNetV2. Numbers above bars indicate total time in ms.

- [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] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
- [7] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018.
- [8] Amazon, "Amazon web services (aws)," <https://aws.amazon.com/>, 2020, accessed: 12 July 2020.
- [9] Intel, "Intel neural compute stick 2," <https://software.intel.com/en-us/neural-compute-stick>, September 2020, accessed: 6 September 2020.
- [10] NVIDIA, "Jetson tx2 module," <https://developer.nvidia.com/embedded/jetson-tx2>, 2020, accessed: 12 July 2020.
- [11] 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.
- [12] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

## References

- [13] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.
- [14] 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.
- [15] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, “Benchmarking keyword spotting efficiency on neuromorphic hardware,” *arXiv preprint arXiv:1812.01739*, 2018.
- [16] 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.
- [17] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikinen, “Deep learning for generic object detection: A survey,” *arXiv preprint arXiv:1809.02165*, 2018.
- [18] 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.
- [19] 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.
- [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] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “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>
- [22] 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.

- [23] D. Pena, A. Foremski, 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.
- [24] 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 *Proceedings of the European Conference on Computer Vision*, 2018, pp. 288–314.
- [25] 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.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.
- [28] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [29] 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.
- [30] 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 *Proceedings of the European conference on computer vision*, 2014, pp. 740–755.
- [31] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010.
- [32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European conference on computer vision*, 2016, pp. 21–37.
- [33] I. Cosmic Shovel, "Amazon price tracker, amazon price history charts, price watches, and price drop alerts." <https://camelcamelcamel.com/>, March 2019, accessed: 24 September 2020.

## References

- [34] 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.
- [35] Intel, “Openvino toolkit,” [https://docs.openvino toolkit.org/2018\\_R5/index.html](https://docs.openvino toolkit.org/2018_R5/index.html), December 2020, accessed: 3 September 2020.

ISSN (online): 2446-1628  
ISBN (online): 978-87-7210-926-8

AALBORG UNIVERSITY PRESS