Anchor tuning in Faster R-CNN for measuring corn silage physical characteristics

Rasmussen, Christoffer Bøgelund; Moeslund, Thomas B.

Published in:
Computers and Electronics in Agriculture

DOI (link to publication from Publisher):
10.1016/j.compag.2021.106344

Publication date:
2021

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

Computers and Electronics in Agriculture 188 (2021) 106344

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Anchor tuning in Faster R-CNN for measuring corn silage physical characteristics☆

Christoffer Bøgelund Rasmussen[a,*], Kristian Kirk[b], Thomas B. Moeslund[a]

[a] *Visual Analysis and Perception Lab Aalborg University, Rendsburggade 14, 9000 Aalborg, Denmark*
[b] *CLAAS E-Systems, Møllevej 11, 2990 Nivå, Denmark*

ARTICLE INFO

Keywords:
Object detection
Deep learning
Corn silage
Kernel fragmentation
Stover

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 lengths 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 (Heinrichs and Coleen, 2016). 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 (Heinrichs and Coleen, 2016). A forage harvester is able to harvest massive amounts of silage in a short period of time and therefore suboptimal machine 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 (Marsh, 2013). 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.

☆ This work was funded by Innovation Fund Denmark under Grant 7038-00170B.

* Corresponding author.
E-mail address: cbra@create.aau.dk (C.B. Rasmussen).
URL: http://www.vap.aau.dk (T.B. Moeslund).

https://doi.org/10.1016/j.compag.2021.106344
Received 16 March 2021; Received in revised form 15 July 2021; Accepted 17 July 2021
Available online 5 August 2021

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) (Heinrichs and Coleen, 2016), 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 (ASABE). 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) (Mertens, 2005) 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 (Mertens, 1997). 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 (Drewry et al., 2019; Luck et al., 2020; Rasmussen and Moeslund, 2019; Rasmussen and Moeslund, 2020). 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 (Savoie et al., 2013; Audy et al., 2014). 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 Rasmussen and Moeslund (2019) 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 Fig. 1 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 overlenths 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 overlengths 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 overlength is 1.5× 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 overlengths 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 overlength 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 overlength object sizes. Our exploration leads us to show that precision and quality measurement can be significantly improved for both overlength and kernel recognition compared to a naive model development approach. Models are evaluated twofold, 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.

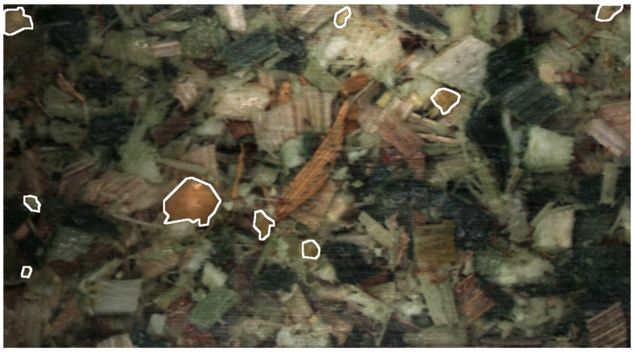


Fig. 1. Example of harvested silage. The white outline shows the kernel fragments and all remaining particles in the image are stover.

Our contribution is threefold:

- 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 (Ren et al., 2015) with an Inceptionv2 (Szegedy et al., 2016) 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 (Kaur and Singh, 2013; Aggarwal and Mohan, 2010; Dalen, 2004; Dubosclard et al., 2015; Guevara-Hernandez and Gomez-Gil, 2011; Patil et al., 2011) or through deep learning systems (Rahnemoonfar and Sheppard, 2017; Bargoti and Underwood, 2017; Sa et al., 2016). 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 (Savoie et al., 2013; Audy et al., 2014), however, both require all particles to be separated and placed in a controlled setting. Within kernel fragmentation, firstly, (Drewry et al., 2019) 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 Luck et al. (2020) the same approach is used to determine the fragmentation of in situ disappearing dry matter. In Rasmussen and Moeslund (2019); Rasmussen and Moeslund (2020), 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.

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 (Igathinathane et al., 2012; Igathinathane et al., 2009; Hamzeloo et al., 2014). Deep learning approaches through CNNs have also been adopted for the task, such as in Frei and Kruis (2019) where a Mask R-CNN was trained to predict the location and classes of agglomerate nanoparticles from which size information could be extracted. In Schäfer et al. (2019) U-Net was adopted for droplet size distribution in chemical engineering applications. In Sharma et al. (2020) 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 (Ren et al., 2015). 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 (Redmon and Farhadi, 2017), 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 Wang et al. (2019) 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 (Zhang et al., 2020) 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 (Yang et al., 2018) 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 Song et al. (2017) and the specific aspect ratio of 0.41 in Zhang et al. (2016). For face detection in Najibi et al. (2017) an aspect ratio of 1:1 was used as faces are generally square in shape. For text detection in Liao et al. (2018) a number of higher and wider aspect ratios were adopted. For ship detection, (Xiao et al., 2019) 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 Rasmussen and Moeslund (2019); Rasmussen and Moeslund (2020) 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 Rasmussen and Moeslund (2019) 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. Fig. 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.

Fig. 3 shows the four class definitions used in this work when differentiating between overlengths. Firstly, we show the classes accepted leaves in Fig. 3 (a) and non-accepted leaves in Fig. 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 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 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 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 standpoint, therefore, these types of leaves are considered ”accepted” as the machine is cutting stover in the manner it is designed to cut. In Fig. 3 (c) we show the class inner-stalk and in Fig. 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 (Garcia, 2020).

Images were annotated producing bounding-boxes for each instance. Table 1 shows an overview of the datasets for the three verbal TLOCs. Firstly, it can be seen that the number of instances per image is greater for a smaller verbal TLOC. As a forage harvester can output hundreds of tonnes per hour this puts a larger requirement on 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 verbal TLOC overlength is at 6 mm compared to a 11.5 mm verbal TLOC at 17.25 mm.

3.1.1. 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 (Mertens, 2005) by sieving a 600 g 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× 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.

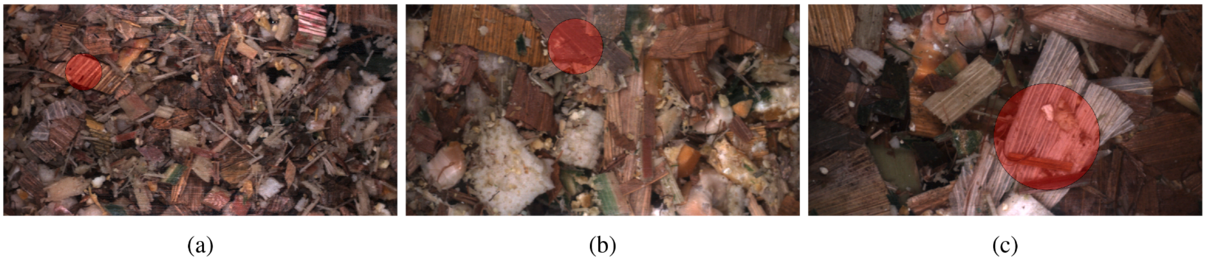


Fig. 2. Example of images from the datasets of different verbal TLOCs. (a) example from *Small* (4 mm). (b) example from *Medium* (6 mm). (c) example from *Large* (11.5 mm). For each image the diameter of the red circle indicates the overlength definition of 1.5× verbal TLOC.

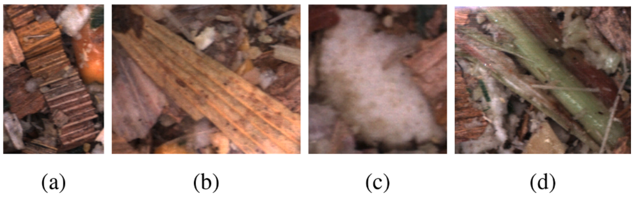


Fig. 3. An example from each of the four classes from silage harvested with an verbal TLOC of 4 mm. (a) accepted leaves, (b) non-accepted leaves, (c) inner stalk, (d) outer stalk.

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 (Huang et al., 2017) 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 × 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 (Lin et al., 2014) 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 × 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 lightweight 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 (Lin et al., 2014) or PASCAL VOC (Everingham et al., 2010) 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 Figs. 4 and 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 Redmon and Farhadi (2017). We see in Fig. 4 that the annotations for kernels have a trend to be slighly longer along their width and sizes largely between 0.1 to 0.2 along both axes. Fig. 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 (Huang et al., 2017) 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 Figs. 4 and 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 Fig. 5 and tune anchors size accordingly further refining the anchor priors and potentially

Table 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).

Verbal 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

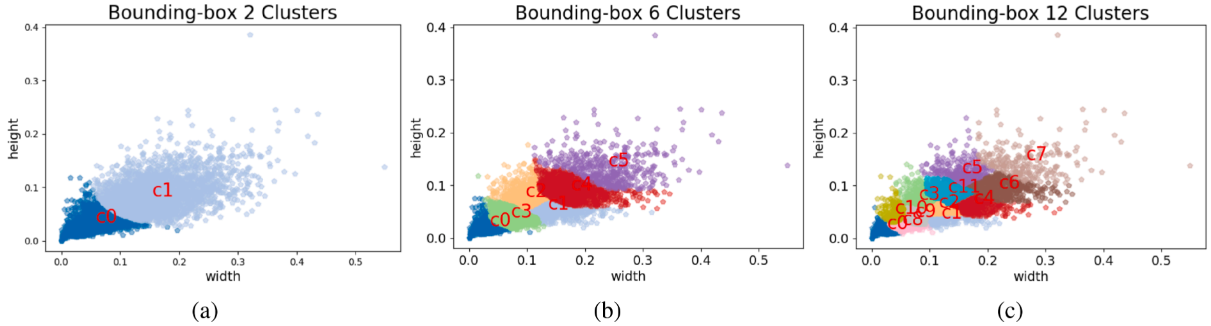


Fig. 4. Anchor height and widths found using k-means clustering for the kernel training set for (a) 2 anchors, (b) 6 anchors and (c) 12 anchors. The anchor centroid determined with k-means is shown by the red text. Each point is the normalised height and width for a bounding-box annotation.

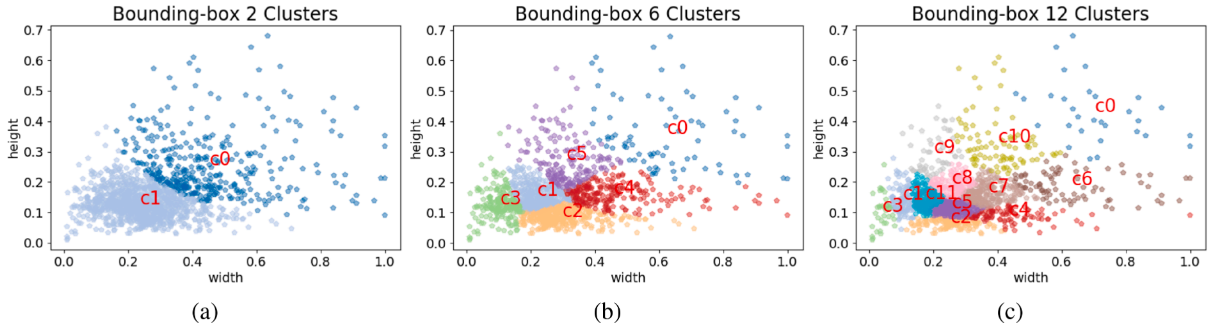


Fig. 5. Anchor height and widths found using k-means clustering for the overlength *All* training set for (a) 2 anchors, (b) 6 anchors and (c) 12 anchors. The anchor centroid determined with k-means is shown by the red text. Each point is the normalised height and width for a bounding-box annotation.

narrow the requirements on RPN optimisation.

3.5. *Converting Predictions to Silage Physical Characteristics 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 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 (Drewry et al., 2019). 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 (Lin et al., 2014) 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 (Everingham et al., 2010) 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 overlength 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 overlengths.

4.1. *Kernel Fragmentation*

For kernel fragmentation we aim to extend our models in comparison to Rasmussen and Moeslund (2019); Rasmussen and Moeslund (2020) by adding anchor tuning to our two-stage networks. In Rasmussen and Moeslund (2020) a Faster R-CNN with Inceptionv2 was trained using a naive strategy with parameters provided from TensorFlow API (Huang et al., 2017) 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 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 3 shows the correlation scores between predicted CSPS based on instances counts below the CSPS threshold and Table 4 for CSPS computed with pixels 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 3 the baseline method performs best. Slight improvements are seen for the week in Table 5 for 2a and 6a. Overall when combining measurements from both harvest weeks the 2a model with CSPS estimated based on pixels has the highest correlation and lowest RMSE.

However, we observed that the Faster R-CNN models lacked small predictions and considerable improvements in PCC and r^2 were made when lowering the confidence threshold. This can be seen in Tables 5 and 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^2 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.

In Fig. 6 we show CSPS estimated with both approaches, first, with instance counts for the baseline scatter plot in 6(a), Faster R-CNN with 2 anchors in 6(b) and Faster R-CNN with 2 anchors at a confidence threshold of 0.05 in 6(c). Secondly, for pixel bounding-boxes for baseline scatter plot in 6(d), Faster R-CNN with 2 anchors in 6(e) and Faster R-CNN with 2 anchors at a confidence threshold of 0.05 in 6(f). In Figs. 6 (c) and 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 Fig. 6 in Fig. 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 7 and Fig. 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, 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 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 (Rasmussen and Moeslund, 2019)	N/A	34.0	NA
MNC (Rasmussen and Moeslund, 2019)	N/A	36.1	NA
Baseline (Rasmussen and Moeslund, 2020)	25.6	51.9	22.3
2a	**28.5**	**56.6**	**25.7**
6a	27.4	55.9	24.0
12a	26.0	54.0	21.0

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 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 9. For both verbal TLOCs the Faster R-CNN trained 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.

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× 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 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 11 shows the effect against the physical samples after filtering

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

	CW40				CW43				CW40 + CW43			
Model	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC
Baseline (Rasmussen and Moeslund, 2020)	0.68	0.46	8.12	0.62	**0.64**	**0.41**	17.09	0.29	0.53	0.28	14.2	0.34
2a	**0.84**	**0.70**	5.39	0.80	0.63	0.40	8.89	0.54	**0.64**	**0.40**	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 4
CSPS estimated with pixel distribution correlation between model estimation and physically measured samples across two harvest weeks.

	CW40				CW43				CW40 + CW43			
Model	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC
Baseline (Rasmussen and Moeslund, 2020)	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	**0.79**	**0.62**	29.59	0.14	0.70	0.48	36.86	0.07	**0.66**	**0.43**	42.44	0.06
6a	0.76	0.58	39.19	0.07	**0.71**	**0.50**	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

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

	CW40				CW43				CW40 + CW43			
Model	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC
Baseline (Rasmussen and Moeslund, 2020)	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	**5.39**	**0.80**	0.63	0.40	8.89	0.54	0.64	0.40	**7.69**	**0.62**
0.25	0.90	0.80	8.90	0.57	0.66	0.44	**7.64**	0.57	0.71	0.51	8.17	0.57
0.05	**0.94**	**0.87**	25.35	0.19	**0.78**	**0.61**	31.3	**0.10**	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	**0.81**	**0.66**	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

Table 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^2	RMSE	CCC	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC
Baseline (Rasmussen and Moeslund, 2020)	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	**0.95**	**0.91**	20.17	0.28	**0.79**	**0.63**	26.32	0.15	**0.81**	0.65	24.05	0.19
0.005	0.92	0.85	16.72	**0.32**	0.77	0.60	20.85	0.20	**0.81**	**0.66**	19.30	0.24
0.0005	0.84	0.71	**16.65**	0.30	0.71	0.51	**18.40**	**0.23**	0.77	0.59	**17.72**	**0.26**

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 10 saw signficant 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.

4.5. Transfer Learning

In Table 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 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^2 improves by 0.23 and 0.32 respectively. Additionally, RMSE decreases to 2.95 and CCC is relatively strong at 0.67.

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 14 provides a summary of the anchor tuning results for the

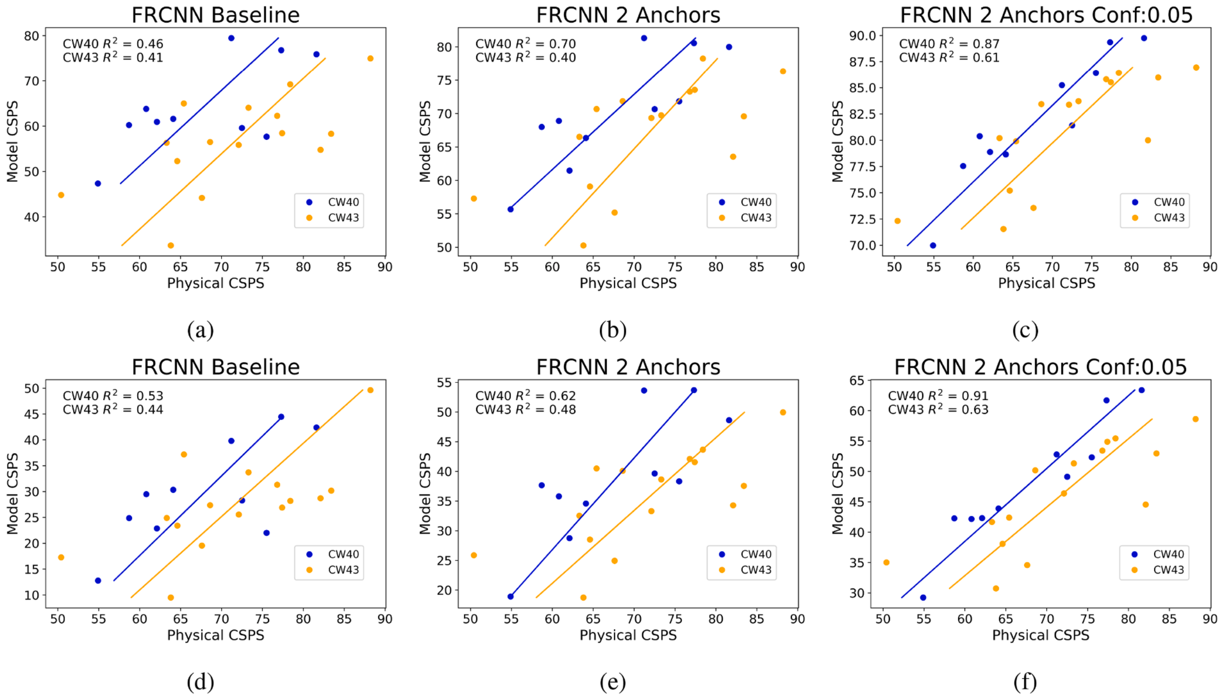


Fig. 6. CSPS estimated with instance counts correlation for the baseline Faster R-CNN (a), Faster R-CNN 2a (b) and Faster R-CNN 2a with a confidence threshold of 0.05 (c). CSPS estimated with pixel distribution for baseline Faster R-CNN (d), Faster R-CNN 2a (e) and Faster R-CNN 2a with a confidence threshold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

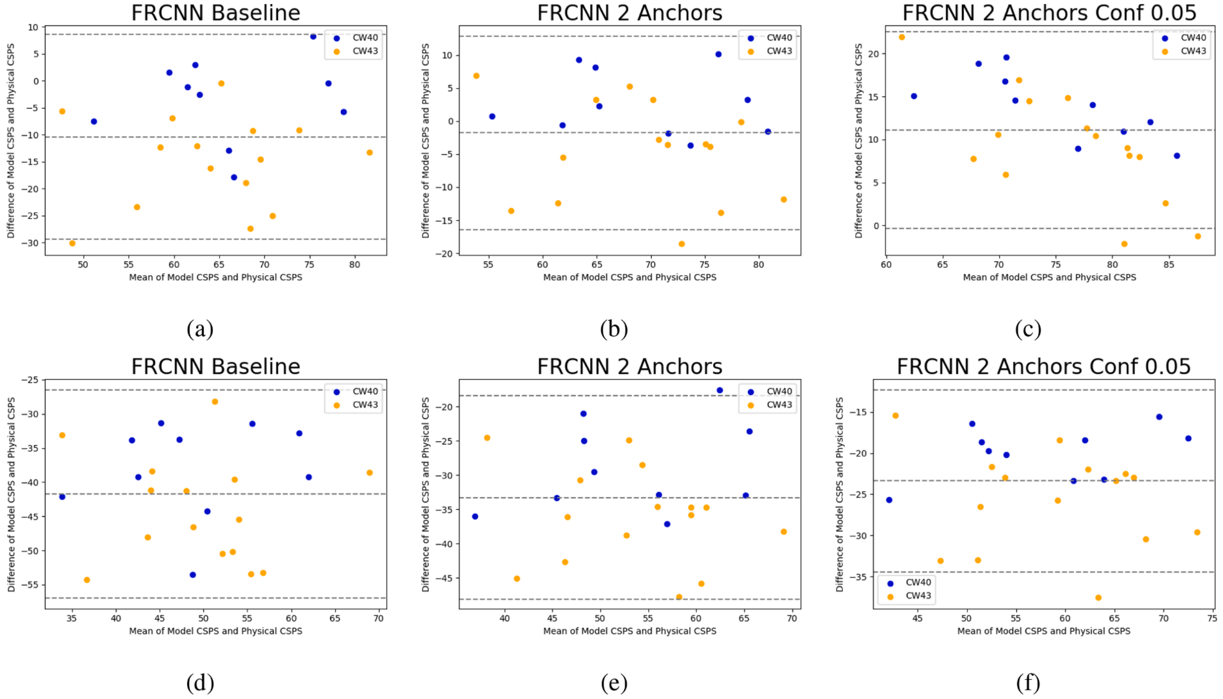


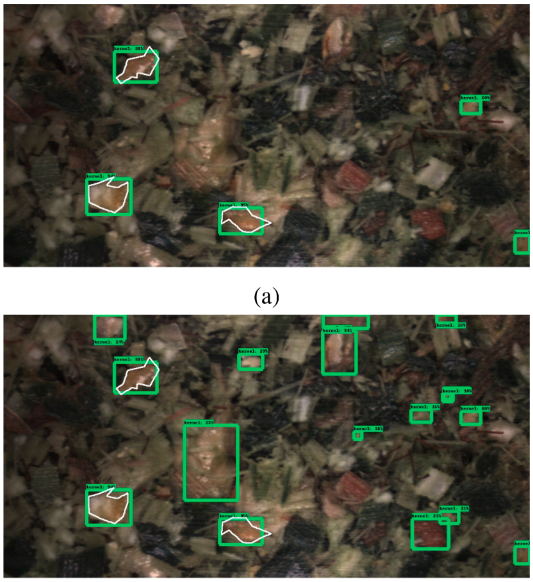
Fig. 7. Bland–Altman plots for the baseline Faster R-CNN (a), Faster R-CNN 2a (b) and Faster R-CNN 2a with a confidence threshold of 0.05 (c). CSPS estimated with pixel distribution for baseline Faster R-CNN (d), Faster R-CNN 2a (e) and Faster R-CNN 2a with a confidence threshold of 0.05 (f).

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*$_{12a}$ increases by 12.9 pp compared to 6.1 pp. Table 14 also

Table 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



(a)

(b)

Fig. 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.

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 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.

In Table 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 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.

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 Fig. 9 are shown in Fig. 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.

Table 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
Small	27.9	52.6	26.1
	-0.1	-0.3	-0.1
Medium	37.8	58.2	46.4
	-0.1	+0.0	+0.0
Large	45.9	62.2	55.5
	+5.4	+9.6	+6.0

Table 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.

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

Table 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
All	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
Small	0.95	0.90	**14.92**	**0.58**	0.41	0.17	51.92	0.02
Medium	**0.97**	**0.93**	31.13	0.17	**0.77**	**0.60**	21.42	0.10
Large	0.86	0.74	35.09	0.07	0.53	0.28	**11.79**	**0.13**

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

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

Table 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
All	28.0	52.9	26.2
Small	32.2	**56.2**	32.3
Small$_{ftall}$	30.1	49.2	**32.6**
All	37.9	58.2	46.4
Medium	36.5	55.2	45.4
Medium$_{ftall}$	**43.3**	**63.0**	**50.4**
All	**40.5**	52.6	**49.5**
Large	27.9	52.6	**49.5**
Large$_{ftall}$	38.6	**55.6**	43.6

Lastly, we show example predictions from the three best performing models for the respective sized test sets in Figs. 11–13. 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.

5. Discussion

5.1. Annotation Data

The increased correlation for CSPS 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

Table 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*.

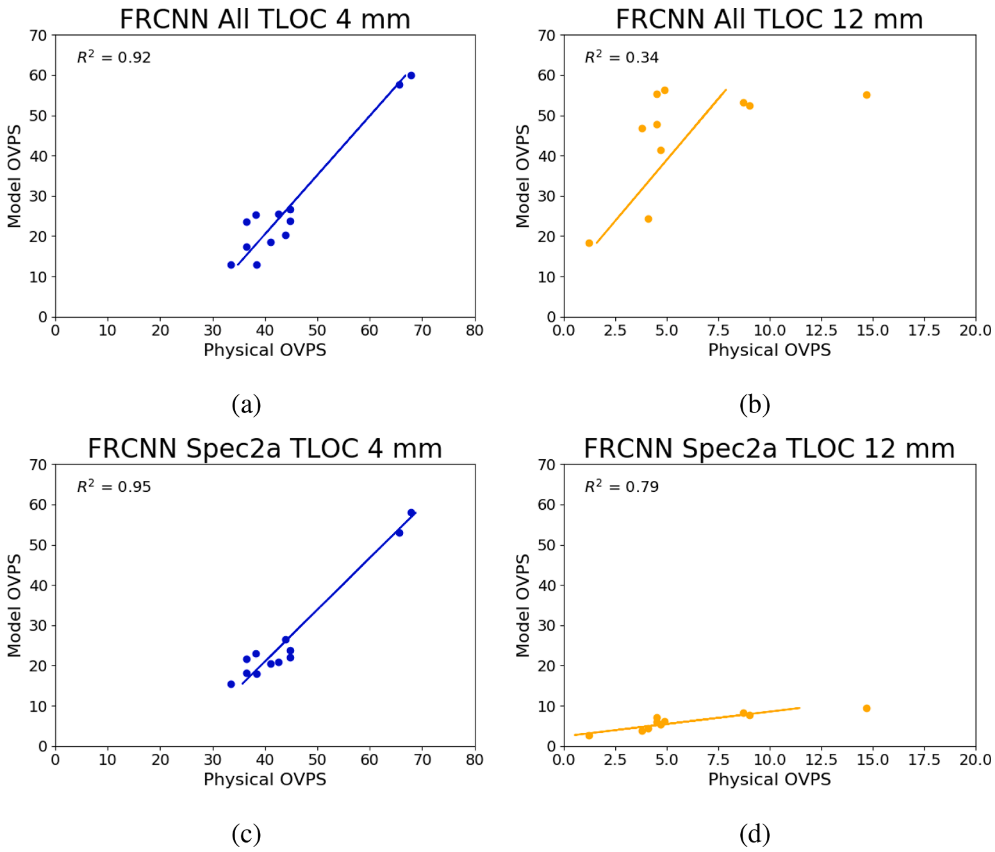
	Verbal TLOC 4 mm				Verbal TLOC 12 mm			
Model	PCC	r^2	RMSE	CCC	PCC	r^2	RMSE	CCC
All	0.96	0.92	17.57	0.49	0.58	0.34	40.63	0.03
Medium$_{coco}$	**0.97**	**0.93**	31.13	0.17	0.77	0.60	21.42	0.10
Medium$_{ftall}$	**0.97**	**0.93**	26.67	0.27	0.63	0.40	29.73	0.06
Resp$_{coco}$	0.95	0.90	**14.12**	**0.58**	0.53	0.28	11.79	0.13
Resp$_{ftall}$	0.96	**0.93**	19.63	0.43	**0.81**	**0.66**	**2.95**	**0.67**

Table 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.

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

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

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

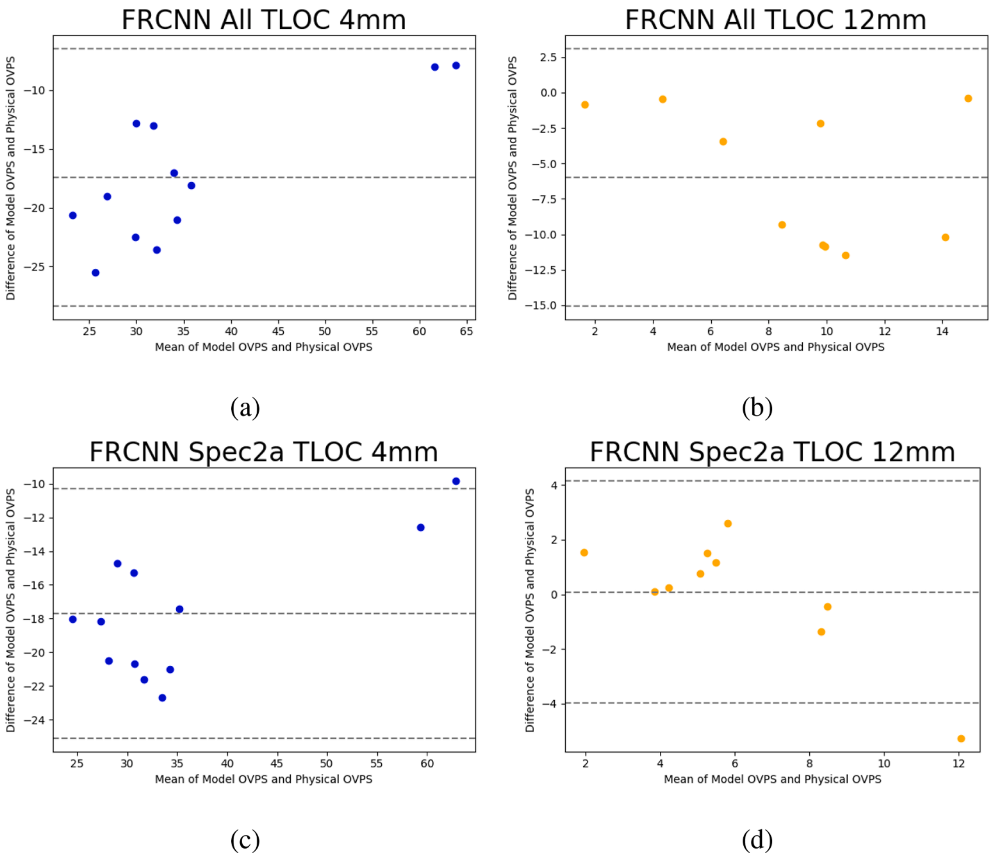


(a) (b)

(c) (d)

Fig. 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).

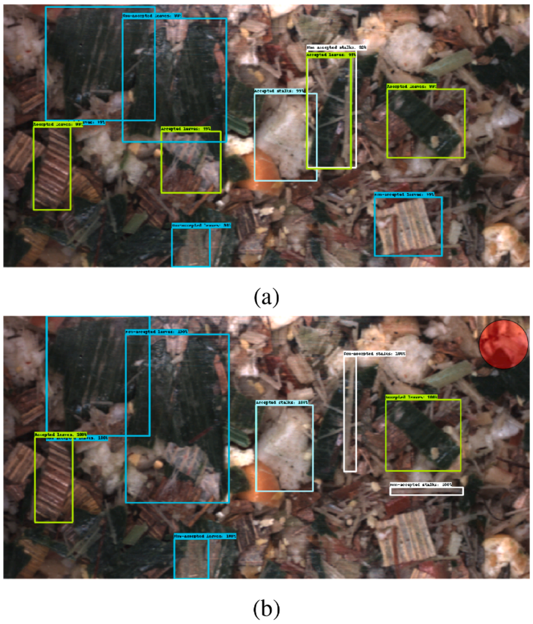
example, in the TensorFlow object detection API (Huang et al., 2017), when visualising the standard threshold is 0.5. However, with a subjective inspection of predictions at such a low threshold there does not appear to be a large number of actual false positives but rather instances that were not annotated. 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



(a) (b)

(c) (d)

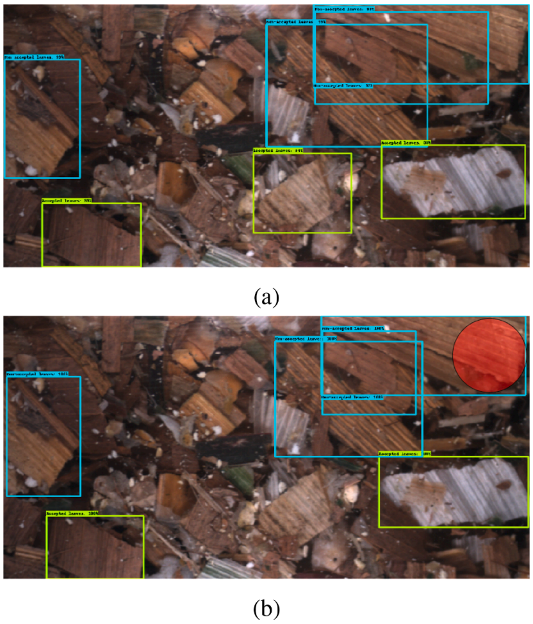
Fig. 10. Bland–Altman plots for different verbal TLOCs. The *All*-trained model at 4 mm (a) and 12 mm (b). Additionally, the best performing model in terms in PCC and r^2 Specialist2a finetuned from *All* at verbal TLOC 4 mm (c) and 12 mm (d).



(a)

(b)

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



(a)

(b)

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

capture the overall particle size distribution and thereby CSPS 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

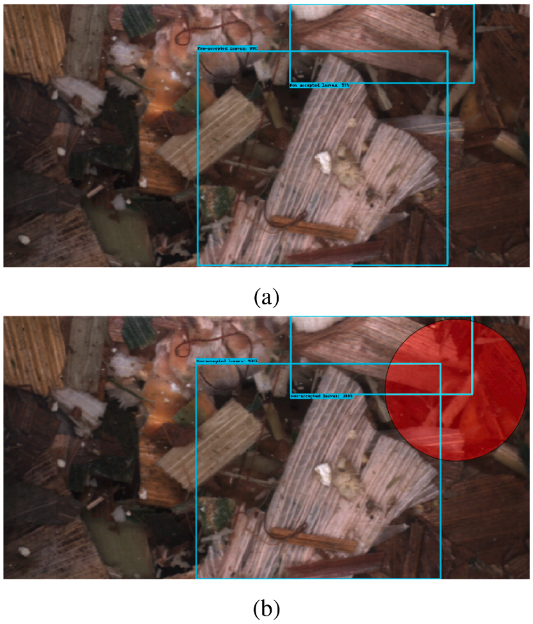


Fig. 13. Example predictions from the Specialist2a finetuned from *All* on images from the *Large* test set. Predictions (a) and ground truth (b).

as Mask R-CNN (He et al., 2017) or with a different backbone feature extractor such as ResNets (He et al., 2016) or MobileNets (Howard et al., 2017). 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. Especially 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 Rasmussen and Moeslund (2020) 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. *Additional Factors Affecting Particle Size*

In this work we have had a focus on evaluating the capability of deep learning models for measuring the physical characteristics of silage. This was done by assessing object recognition metrics against annotations and a statistical analysis of physically sieved samples. Within the latter and when creating our image sets sample points were grouped based on the verbal TLOC and processor gap in order to cover varying particle sizes. However, there are a large number of additional factors which we have not considered in both the analysis or when designing models. This can include the maturity of the plant, geographical differences, or the efficacy of the forage harvester. Our models have shown a promising result but is largely centred around variations in the harvester settings. Therefore, future work should assess the models across a larger dataset covering more variation. Based upon these results the validity of our approach will be tested to see how much they generalise. Possibly larger annotated training datasets are required or even more specialised models for specific factors.

5.4. *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 Fig. 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 Fig. 14 the CSPS calculatation 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.

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 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

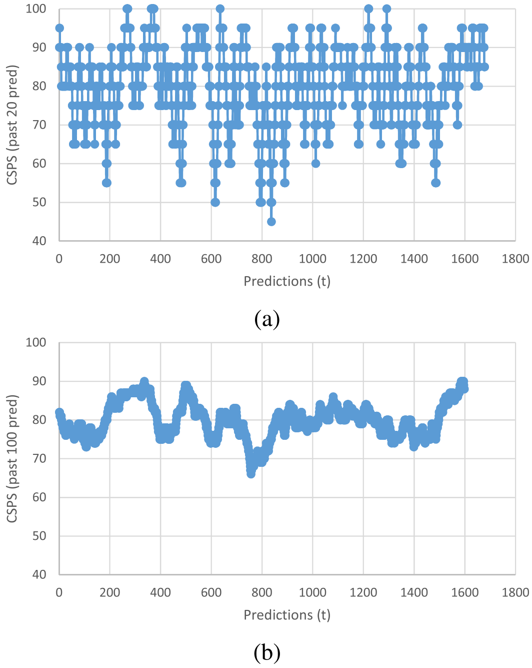


Fig. 14. Rolling average for CSPS for an image set for the past 20 predictions (a) and past 100 predictions (b). The best performing model found in this work computed a CSPS of 80.38 over the entire image set.

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.

CRediT authorship contribution statement

Christoffer Bøgelund Rasmussen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Funding acquisition. **Kristian Kirk**: Conceptualization, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Thomas B. Moeslund**: Conceptualization, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aggarwal, A., Mohan, R., 2010. Aspect ratio analysis using image processing for rice grain quality. Int. J. Food Eng. 6 https://doi.org/10.2202/1556-3758.1788.

American Society of Agricultural and Biological Engineers(ASABE), Method of determining and expressing particle size of chopped forage materials by screening. ANSI/ASAE S424.1, 663–665.

Audy, M., Savoie, P., Thibodeau, F., Morissette, R., 2014. Size and shape of forage particles by image analysis and normalized multiscale bending energy method. In: American Society of Agricultural and Biological Engineers Annual International Meeting 2014, ASABE 2014 2, 820–830.

Bargoti, S., Underwood, J., 2017. Deep fruit detection in orchards, pp. 3626–3633. https://doi.org/10.1109/ICRA.2017.7989417.

Dalen, G., 2004. Determination of the size distribution and percentage of broken kernels of rice using flatbed scanning and image analysis. Food Res. Int. 37, 51–58. https://doi.org/10.1016/j.foodres.2003.09.001.

Drewry, J.L., Luck, B.D., Willett, R.M., Rocha, E.M., Harmon, J.D., 2019. Predicting kernel processing score of harvested and processed corn silage via image processing techniques. Computers and Electronics in Agriculture 160, 144–152. URL: http://www.sciencedirect.com/science/article/pii/S0168169918311955, doi: 10.1016/j.compag.2019.03.020.

Dubosclard, P., Larnier, S., Konik, H., Herbulot, A., Devy, M., 2015. Automatic visual grading of grain products by machine vision. J. Electron. Imaging 24, 061116. https://doi.org/10.1117/1.JEI.24.6.061116.

Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge.

Frei, M., Kruis, F., 2019. Image-based size analysis of agglomerated and partially sintered particles via convolutional neural networks. Powder Technol. 360 https://doi.org/10.1016/j.powtec.2019.10.020.

Garcia, A., 2020. Corn silage production and utilization. URL: https://extension.sdstate.edu/sites/default/files/2019-09/S-0003-18-Corn.pdf.

Guevara-Hernandez, F., Gomez-Gil, J., 2011. A machine vision system for classification of wheat and barley grain kernels. Spanish Journal of Agricultural Research 9, 672. https://doi.org/10.5424/sjar/20110903-140-10.

Hamzeloo, E., Massinaei, M., Mehrshad, N., 2014. Estimation of particle size distribution on an industrial conveyor belt using image analysis and neural networks. Powder Technol. 261, 185–190. https://doi.org/10.1016/j.powtec.2014.04.038.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Heinrichs, J., Coleen, M.J., 2016. Penn state particle separator. URL: https://extension.psu.edu/penn-state-particle-separator.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. URL: http://arxiv.org/abs/1704.04861. cite arxiv:1704.04861.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In: in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3296–3297. https://doi.org/10.1109/CVPR.2017.351.

Igathinathane, C., Pordesimo, L., Columbus, E., Batchelor, W., Sokhansanj, S., 2009. Sieveless particle size distribution analysis of particulate materials through computer vision. Computers and Electronics in Agriculture - COMPUT ELECTRON AGRIC 66, 147–158. https://doi.org/10.1016/j.compag.2009.01.005.

Igathinathane, C., Ulusoy, U., Pordesimo, L., 2012. Comparison of particle size distribution of celestite mineral by machine vision volume approach and mechanical sieving. Powder Technology - POWDER TECHNOL 215–16. https://doi.org/10.1016/j.powtec.2011.09.037.

Kaur, H., Singh, B., 2013. Classification and grading rice using multi-class svm. International Journal of Scientific and Research Publications 3, 1–5.

Liao, M., Shi, B., Bai, X., 2018. Textboxes++: A single-shot oriented scene text detector. IEEE Trans. Image Process. pp. https://doi.org/10.1109/TIP.2018.2825107.

Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. ECCV 740–755.

Luck, B.D., Drewry, J.L., Shaver, R.D., Willett, R.M., Ferraretto, L.F., 2020. Predicting in situ dry matter disappearance of chopped and processed corn kernels using image-analysis techniques. Applied Animal Science 36, 480–488. URL: https://www.sciencedirect.com/science/article/pii/S2590286520300847, doi:https://doi.org/10.15232/aas.2020-01993.

Marsh, B.H., 2013. A comparison of fuel usage and harvest capacity in self-propelled forage harvesters. International Journal of Agricultural and Biosystems Engineering 7, 649–654. URL https://publications.waset.org/vol/79.

Mertens, D., 1997. Creating a system for meeting the fiber requirements of dairy cows. Journal of Dairy Science 80, 1463–1481. URL: https://www.sciencedirect.com/science/article/pii/S0022030297760752, doi: 10.3168/jds.S0022-0302(97)76075-2.

Mertens, D., 2005. Particle size, fragmentation index, and effective fiber: Tools for evaluating the physical attributes of corn silages. In: In: Proceedings of the Four-State Dairy Nutrition and Management Conference.

Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S., 2017. SSH: single stage headless face detector, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017, IEEE Computer Society. pp. 4885–4894. URL: doi: 10.1109/ICCV.2017.522, doi:10.1109/ICCV.2017.522.

Patil, N., Malemath, V., Yadahalli, R.M., 2011. Color and texture based identification and classification of food grains using different color models and haralick features. International Journal on Computer Science and Engineering 3.

Rahnemoonfar, M., Sheppard, C., 2017. Deep count: Fruit counting based on deep simulated learning. Sensors (Basel, Switzerland) 17. https://doi.org/10.3390/s17040905.

Rasmussen, C., Moeslund, T.B., 2020. Evaluation of model selection for kernel fragment recognition in corn silage. arXiv:2004.00292.

Rasmussen, C.B., Moeslund, T.B., 2019. Maize silage kernel fragment estimation using deep learning-based object recognition in non-separated kernel/stover rgb images. Sensors 19, 3506. https://doi.org/10.3390/s19163506.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 28. Curran Associates Inc, pp. 91–99. URL: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., Mccool, C., 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors 16, 1222. https://doi.org/10.3390/s16081222.

Savoie, P., Audy-Dubé, M.A., Pilon, G., Morissette, R., 2013. Chopped forage particle size analysis in one, two and three dimensions. doi:10.13031/aim.20131620148.

Schäfer, J., Schmitt, P., Hlawitschka, M.W., Bart, H.J., 2019. Measuring particle size distributions in multiphase flows using a convolutional neural network. Chem. Ing. Tech. 91, 1688–1695. https://doi.org/10.1002/cite.201900099. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.201900099, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201900099.

Sharma, K., Gold, M., Zurbrügg, C., Leal-Taixé, L., Wegner, J., 2020. Histonet: Predicting size histograms of object instances, pp. 3626–3634. doi:10.1109/WACV45572.2020.9093484.

Song, D., Qiao, Y., Corbetta, A., 2017. Depth driven people counting using deep region proposal network, pp. 416–421. https://doi.org/10.1109/ICInfA.2017.8078944.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. https://doi.org/10.1109/CVPR.2016.308.

Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D., 2019. Region proposal by guided anchoring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Xiao, X., Zhou, Z., Wang, B., Li, L., Miao, L., 2019. Ship detection under complex backgrounds based on accurate rotated anchor boxes from paired semantic segmentation. Remote Sensing 11, 2506. https://doi.org/10.3390/rs11212506.

Yang, T., Zhang, X., Li, Z., Zhang, W., Sun, J., 2018. Metaanchor: Learning to detect objects with customized anchors. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31. Curran Associates Inc, pp. 320–330. URL: http://papers.nips.

cc/paper/7315-metaanchor-learning-to-detect-objects-with-customized-anchors.pdf.

Zhang, L., Lin, L., Liang, X., He, K., 2016. Is faster R-CNN doing well for pedestrian detection?, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision - ECCV 2016–14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II, Springer. pp. 443–457. URL: doi: 10.1007/978-3-319-46475-6_28, doi:10.1007/978-3-319-46475-6_28.

Zhang, S., Wen, L., Lei, Z., Li, S.Z., 2020. Refinedet++: Single-shot refinement neural network for object detection. IEEE Trans. Circuits Syst. Video Technol. 1–1.