Enhancing Sheep Detection Techniques: Final Report

G002 (s2096890, s2107370, s2101906)

Abstract

Sheep counting in farming is traditionally done manually, making it a time-consuming and error-prone task. Due to deep learning techniques, non-invasive approaches have been introduced to automate this process, however, many challenges persist when considering occlusion between sheep and the limited data available.

Our work focuses on extending these applications to real-world scenarios and exploring the impact of behaviour and movement on detection accuracy. We utilise a new dataset that captures sheep in different settings and backgrounds, using labels that showcase each sheep's activity and taking into consideration whether there is occlusion. We demonstrate which model is best for the object detection task at hand, which is the Single Shot Detector (SSD). We tested the effects of applying transfer learning to the model which gave poor results. Consequently, we fine-tuned our model with techniques like data augmentation and the Adam optimiser, to improve the performance of our model.

Overall, the study highlights the complexity of automating sheep counting using deep learning methods and shows the importance of considering diverse environmental conditions and sheep behaviours for improved model accuracy.

1. Introduction

Sheep counting is an uphill task that is done manually in farms around the world. In the US, for instance, shepherds handle thousands of sheep throughout the summer, incapable of making accurate counts (Wollan, 2017). It is a task of utmost importance, especially since sheep rustling has emerged with incidents costing over £2.5 million due to stolen sheep in 2018 (Morris, 2020). Therefore, making regular livestock inventories is essential, not only to detect theft and escapes but to effectively divide grassland resources. Without an accurate and efficient method for counting sheep, the grazing intensity cannot be controlled properly, which can lead to overgrazing: sheep being overfed (Xu et al., 2022).

Traditional, manual counting is not only inefficient but error-prone due to duplication and omission of sheep. This is due to the nature of sheep to cluster together, producing mutual occlusions. Other present techniques use electronic ear tags and, even though they prove to be accurate, they are too expensive for large-scale farms and can physically damage the animals, causing infections. Consequently, due to the ongoing development of machine learning, deep-learning techniques have been applied to the problem at hand as a non-invasive approach.

The main goal of our work is to expand the research on this real-world application. Existing experiments have overlooked the usage of videos and their frames with illumination and angle variations, and background diversity. In addition, they have failed to consider the various behaviours of sheep, often assuming a static stance or walking. It is our task to produce accurate detection for sheep from a newly created dataset, that takes into consideration the factors that dropped the accuracy in previous models. For this purpose, we labelled each sheep with their corresponding behaviour in order to discover whether sheep who are standing, walking, running, sitting or grazing affect the model's accuracy. This could be of special interest to farmers, as it could affect the placement of surveillance cameras on their farms.

The state-of-the-art model for real-time detection involves a Single-Shot Detector (SSD), which is why it was selected as the model to work with in this investigation. Video processing and handling is a heavy task to work on, and therefore, considering our limited computing resources, we designed our approach to emphasise the use of transfer learning. This enables us to leverage a pretrained model as a baseline and fine-tune it by training it further with our dataset. Hence, it minimises the need for substantial data while still obtaining meaningful results.

As a result, our objective is to explore the effects of applying transfer learning with a diverse dataset to the task, aiming to improve the limitations found in the relevant literature. With training data which includes diverse backgrounds, different sheep activities and overlapped bodies, we aim to achieve a better model for accurate sheep counting, in order to assist farmers in addressing their sheep counting challenges.

In the following section, we describe the relevant work related to sheep counting and detection, to emphasise how this project distinguishes from them. In Section 3 we describe the datasets used in our experiments, showing how they were preprocessed and how our model is trained and evaluated. Section 4 gives an overview of how the SSD model was used and how we extended it, and Section 5 shows the results of the experiments done with this model. Further analysis of occlusion and sheep activities can be found in Section 6 and, finally, in Section 7 we provide the

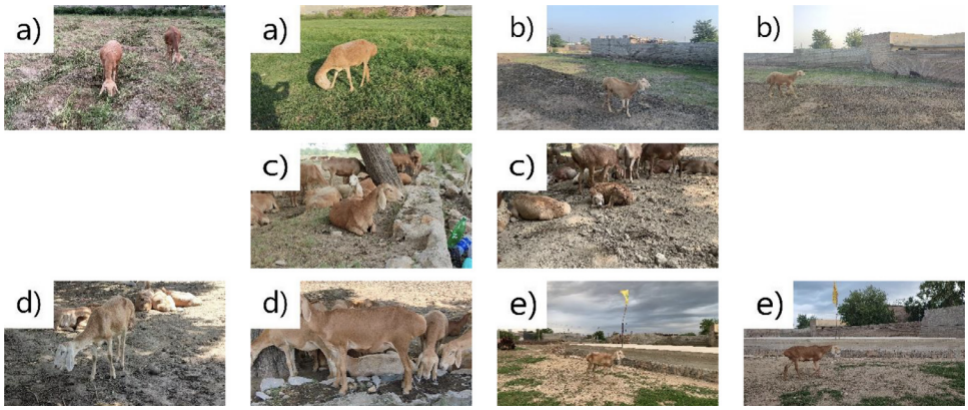

Figure 1. Frames which represent the five types of activities: a) grazing, b) running, c) sitting, d) standing, and e) walking from (Kelly et al., 2024)

key outcomes of the investigation, with potential suggestions for improvements.

2. Related work

Limited research has been carried out in object detection specific to sheep and farming, but, within the existing investigations, the extensive majority have been carried out with Unmanned Aerial Vehicles (UAV) images. Deep Convolutional Neural Networks (CNNs) are well-known for problems concerning detection, localization and classification, and thus these have been applied and described in several papers. For instance, methods involving R-CNN have been used for sheep detection (Sarwar et al., 2018) (Sarwar et al., 2021). However, limitations were found since the images were taken under varying elevations and weather, meaning that illumination and sheep scales were not constant which hindered the accuracy.

Nonetheless, images done with UAVs are not considered practical since they do not allow regular sheep counting and are hard to take inside farms. A suitable solution for this entails using videos, which can be collected from surveillance cameras used to control the sheep. Crow counting methods have been employed (Xu et al., 2022) (Yu et al., 2023), which resulted in high accuracies, especially in varying crowd densities and distributions. Nonetheless, this technique focuses on estimating the total sheep amount and therefore, does not provide individual object IDs. Farmers who need detailed information about individual sheep, such as tracking and behaviour analysis, might find this method impractical.

Relevant work has been done in order to accurately automate sheep tracking and counting within practical farming environments, by utilising suitable object detection and tracking techniques. To do this, data was collected from a passage where sheep entered and exited (Wang et al., 2023), updating the count when the sheep went through a threshold. Here, an adaptation of the SSD technique was employed for recognition, which is considered the best model for real-time detection due to a good balance in velocity and accuracy. However, the dataset used was limited and did not contain a variation of backgrounds, meaning that the model cannot be used in diverse environments. Additionally, although the model had high overall accuracies, these dropped when there was a high density of sheep, which was a result of occlusion.

Therefore, even though research has explored deep learning techniques for sheep detection, it has encountered many limitations. UAVs are impractical in carrying out regular counting procedures, which shows the value of exploring video-based solutions. However, the relevant work that tried these solutions did not take into consideration different sheep activities that could affect the accuracy of the model and faced challenges when dealing with occlusion.

3. Data set and task

3.1. Dataset Description

The data used in the project was obtained from the Video Dataset of Sheep Activity (Khan & Kelly, 2023), published in October of 2023. Here, sheep were recorded during different activities including grazing, running, walking, standing and sitting with a good variation of angles and backgrounds, as seen in Figure 1. It contains a total of 149,327 frames from 417 videos that are on average 8.5 seconds long, which are proven useful for recognition and detection models. The videos were taken with two devices, an iPhone XS Max which records 60 frames per second (FPS) at 4K, and a Redmi Note 10 Pro, recording 30 FPS at 4K.

It is considered to be the first sheep dataset containing this kind of diverse data, which can be useful for real-time sheep management (Kelly et al., 2024). The dataset is labelled for each type of activity, giving us the opportunity to test which behaviour gives the best detection accuracy and, thus, a more precise sheep count.

Keeping in mind the scarce computing resources that were available, we decided to use transfer learning. Therefore, we use a pretrained model on the Common Objects in

Context (COCO) dataset (Lin et al., 2014). It is a well-known large-scale image recognition dataset, that contains over 330,000 images and 91 labels, including sheep. This dataset has been used extensively in computer vision research, specifically for object detection, which is why we made use of it for our pretrained model. This will also allow us to create baseline experiments in order to corroborate whether transfer learning will improve the accuracy of the model when trained on a dataset, which includes videos with occlusion and different sheep activities.

3.2. Data Preprocessing

To test our main goal of improving detection for sheep counting, we need each frame to contain bounding boxes confining each sheep. Consequently, preprocessing is necessary to obtain this information since the dataset does not contain these labels. We had to manually go through the frames and label them accordingly before any training could be done.

Each frame from every video was assigned a bounding box for each sheep, and labelled with their corresponding behaviour. In addition, we labelled the videos with an occlusion tag when sheep were interposing themselves, allowing us to discover whether our model improved its accuracy when there were occluding sheep. After filtering the dataset and omitting videos in miscellaneous categories, we reduced it to a total of 318 labelled videos, which have 1,899 ground truth annotations, depicted in Table 1.

Label	Number of annotations
Sheep Grazing	559
Sheep Sitting	619
Sheep Standing	261
Sheep Running	63
Sheep Walking	79
With Occlusion	144
Without Occlusion	168

Table 1. Labels used during preprocessing with the total annotations done

In response to the lack of data concerning sheep detection and their behaviours, we established a repository containing the videos along the labels and annotations, as a contribution to the scientific community. Due to our computing resource constraints, we were unable to make use of the entire preprocessed dataset, prompting us to share it for potential future investigations by others. Consequently, to train our model we used 50 of the 318 labelled videos, where we ensured an equal distribution across the different sheep activities. For validation and testing purposes, we approached an 80-10-10 split, which is recommended for good practice.

3.3. Task Evaluation

For sheep detection, the standard object detection metric involves Precision (1), to measure the accuracy of the positive predictions, and Recall (2), to measure the ability of the model to capture all positive instances. To obtain the True Positive (TP), False Positive (FP) and False Negative (FN) counts we compute the Intersection over Union (IoU) (3), which measures the spatial overlap between the predicted bounding boxes and the ground truths.

Then, to assess the accuracy of the different sheep activities, we calculate the recall for every category. We have no FP scores for the activity classes as we are not doing classification, therefore, it is not possible to calculate the precision in this instance.

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

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

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

With these metrics, we can calculate the F1 score (4) to evaluate the overall performance, since it is the harmonic mean of precision and recall. Hence, we provide a single score that can balance both measures.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

With the Mean Absolute Value (MAE), we can calculate the absolute differences between the predicted and the actual values (5), concerning the number of bounding boxes in each frame, where D is the total number of frames. Although this will give us a more intuitive understanding of the average error magnitude, it will treat large and small errors the same, which is why we introduce the Mean Squared Error (MSE) (6). This way, we obtain the average of the squared differences between the predicted counts and the ground truth, where large errors will be penalised more.

$$MAE = \frac{1}{D}\sum_{i=1}^{D}(x_i - y_i)^2 \tag{5}$$

$$MSE = \frac{1}{D}\sum_{i=1}^{D}|(x_i - y_i)| \tag{6}$$

4. Methodology

The main goal of this project is object detection, a task that has been deeply explored in the field of computer vision over the past years. As it has been previously mentioned, this is a rather challenging task when occlusions and diverse scenarios are present.

Given the complexity of the task, we decided to implement transfer learning on a pretrained model, due to the reduced

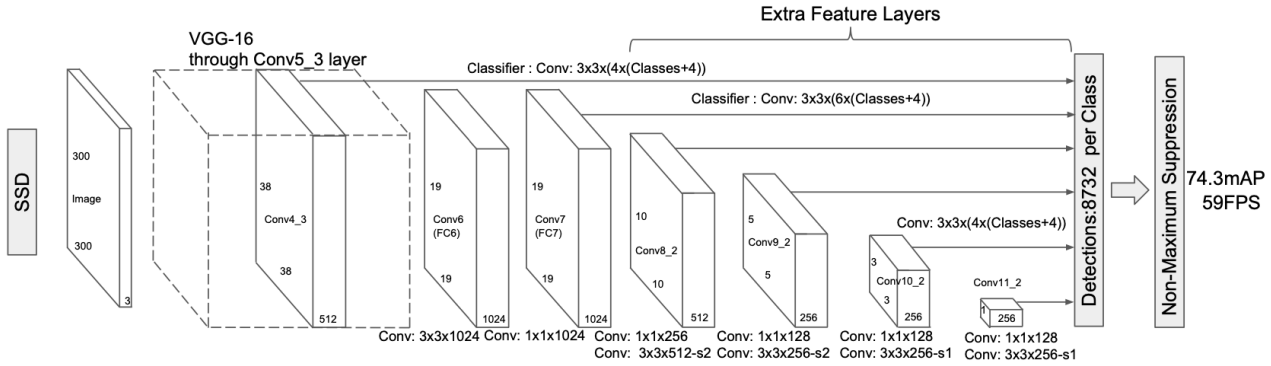

Figure 2. Representation of the SSD300-VGG16 model

need for data resources for training that this technique offers. After thorough research about the state-of-the-art methods, it was decided that the best model choice was the Single Shot Detector (SSD), given its good performance handling occlusion in previous research on sheep counting, as opposed to other models. Even though this is a simple method, it has been proven to be not only fast, but also very accurate.

More specifically, we used the SSD300-VGG16 model trained on the COCO dataset. The use of a pretrained model offers several advantages, the main one being feature reuse. The pretrained model has already learnt to extract useful features from images. This includes specific patterns or structures which will be very useful for our task: sheep detection.

We will use the weights of the pretrained model to initialize the weights of our own model. Then, through transfer learning, we will re-train the model to adjust it to better fit our specific task. This entails using the knowledge that is stored in the pretrained model to accelerate the convergence of the model during training. Moreover, this will also significantly reduce the computational resources needed as the model is not learning from scratch but leveraging already existing knowledge. For this reason, the amount of training data needed will be significantly reduced, as our model will use the features learnt from a larger dataset to improve the performance on our smaller dataset.

Finally, a pretrained model that has been trained on a large dataset such as COCO will likely generalize better to unseen data, and therefore will potentially improve the performance of our model.

Focusing now on the choice of the SSD model, which outperforms the extensively used Faster R-CNN. The latter is based on: hypothesising bounding boxes, resampling pixels or features for each box, and applying a high-quality classifier (Liu et al., 2016). This approach has been established to be computationally expensive and too slow for real-life applications. The detection speed of these types of methods is normally measured in FPS, and Faster R-CNN only managed to operate at 7 FPS. The ideal FPS for video analytics should be between 10 and 30 frames, so this represented a problem for our project as our dataset only contains videos (Liu et al., 2016). For this reason, we discarded this type of method and decided to rely on SSD.

The Single Shot Detector discretized the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. After that, during the prediction, thanks to the feed-forward convolutional network, it will generate scores for the presence of each object category in each of those default boxes. It will also make adjustments with the goal of better matching the object shape. Not only that, but the network will also handle objects of multiple sizes by combining predictions from feature maps with different resolutions. This is especially important taking into account that in the videos contained in the dataset, the sheep might be moving and, therefore, changing their position with respect to the camera.

The SSD300-VGG16 model includes as a base network the well-known VGG16 architecture. Multiple extra layers are added to predict the offsets of the default boxes of different scales and aspect ratios, as well as the associated confidences. The model is shown in Figure 2. The VGG16 and the extra layers constitute the two main components of the model: the backbone and the head, respectively. The backbone is in charge of extracting features from the input image and for our specific model, it is derived from a VGG16 model. The VGG16 model, a convolutional neural network architecture, is extensively used in computer vision tasks for its simplicity and uniform structure (Qassim et al., 2018).

The backbone also determines the quality of these extracted features, a key step for the subsequent detection performance. On the other hand, the head of the model makes the predictions based on the features extracted by the backbone. It is in charge of predicting the bounding boxes, as well as the class scores of potential objects in the image. During the training of the head of the model, the loss is computed by comparing the predictions with the ground truth labels.

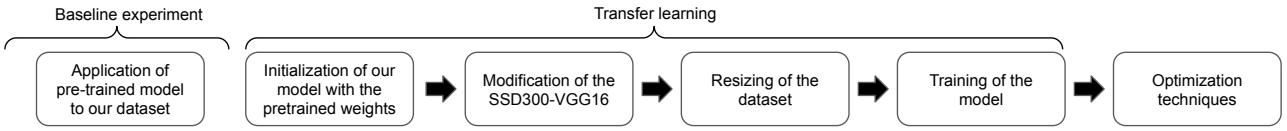

Figure 3. Pipeline of the project

However, for the purpose of our transfer learning experiments, we modified the structure of the model to adapt it to the sheep detection task. We froze the weights from the backbone of the model and, therefore, only the weights from the head were trained and updated. Furthermore, and keeping in mind that the COCO dataset includes 91 different classes, we modified the classification head so that it could only output two classes: sheep and background.

The remaining major modification needed was the adaption of our dataset to the model. The input size of the images for SSD300-VGG16 is 300x300 (Liu et al., 2016). However, the videos from our dataset were of two different sizes: 3820x2160 or 1920x1080, therefore they had to be resized, along with the bounding boxes, so that we could successfully implement transfer learning. Further minor modifications were carried out to adapt the images to the corresponding compatible data types.

Once the model was trained and we obtained a baseline for its performance, multiple combinations of optimization techniques were tried. Firstly, we used Stochastic Gradient Descent (SGD). Traditional gradient descent aims to find the model parameters that will minimize the error on the entire dataset. For this variation, SGD, instead of using the whole dataset to compute the gradient, will only use one single random data point to update the parameters (Hardt et al., 2016). Alternatively, some of our experiments replace SGD with the Adam optimizer, to explore their different effects. The Adam optimizer combines two other optimization algorithms which are: Adaptive Gradient Algorithm (AdaGrad), focused on problems with sparse gradients, and Root Mean Square Propagation (RMSProp), which will handle non-stationary problems. The Adam optimizer keeps two moving averages for each parameter, one for the gradients and another one for the squared gradients. This will help to dynamically update the learning rate of each parameter (Bock et al., 2018).

The pipeline of the whole process can be visualized in Figure 3. First, we applied the pretrained model to our database in order to obtain a baseline on the performance. The weights of the pretrained model were saved as they need to be used later to initialise our model during transfer learning. After this, the SSD300-VGG16 model was modified, as was previously mentioned, so it suits better our goal task. In order to match the input size of the model, our dataset and labels were resized so that the videos were of dimensions 300x300. Finally, we will re-train our model.

5. Experiments

5.1. SSD vs. R-CNN

Description of the experiment: The main goal of the experiment is a comparison between a pretrained SSD model and a pretrained Faster R-CNN model. Their performance was compared across different thresholds for the task of detection.

Experimental design: The experiments were carried out on a reduced dataset of 203 images (Ninja, 2024) that were labelled with their corresponding bounding boxes. This dataset is representative of data previously used to train sheep detectors, including simplistic pictures with limited occlusion and diversity of backgrounds. The experiment was focused on the task of object detection, as it is a crucial part of our project to corroborate the superiority of the SSD while using different thresholds. We chose the values 0.1, 0.4 and 0.7, meaning that the model will only consider the bounding boxes that the models predicted with a confidence above that specified threshold.

Results: In general, and for the purpose of comparison, the better model should have a higher accuracy, precision, recall and F1 score as this will indicate better performance. However, for the MAE and MSE, as they are metric errors, ideally we would want them to be as low as possible.

The results for both models are recorded in Table 2. Let's do an individual analysis for each of the thresholds. For the 0.1 threshold, the SSD model presents better results for MAE, MSE, accuracy, precision and F1 score than the Faster R-CNN. However, the latter obtained a slightly higher value for recall. When the threshold is 0.4, once again, the SSD model outperformed Faster R-CNN in all metrics except in recall, for which the latter obtained a value of 0.99. Finally, when the threshold was 0.7 the SSD model obtained more favorable results for all the metrics except for recall. In general, it is clear then that the SSD300-VGG16 seems to perform better for the task of object detection.

Conclusions: The SSD300-VGG16 will consistently have a lower MAE and MSE which indicates a better prediction accuracy and fewer errors. Moreover, this model obtained higher values for accuracy and F1 scores, which implies that the model does not only make correct predictions more often but also maintains a balance between precision and recall. It will be better at identifying true positives and negatives.

The fact that the Faster R-CNN obtains higher recall implies that it is better at identifying relevant instances, but this will

Model	Threshold	MAE	MSE	Acc.	Precision	Recall	F_1
SDD_300_VGG16	0.1	0.75	133.81	0.68	0.69	0.98	0.81
fasterrcnn_resnet50	0.1	3.23	2113.42	0.26	0.34	1.0	0.50
SDD_300_VGG16	0.4	0.16	5.36	0.90	0.95	0.95	0.95
fasterrcnn_resnet50	0.4	0.87	154.33	0.56	0.65	0.99	0.79
SDD_300_VGG16	0.7	0.22	9.54	0.86	0.98	0.89	0.93
fasterrcnn_resnet50	0.7	0.43	38.15	0.72	0.80	0.97	0.88

Table 2. Experiment 5.1 results

be at the cost of high error rates and lower precision. This will lead to a higher number of false positives.

In conclusion, the SSD300-VGG16 exhibited a superior performance across different thresholds on most metrics and, therefore, was the chosen model to perform object detection.

5.2. Baseline Experiment with SSD300-VGG16

Description of the experiment: Initially, we test our dataset on the pretrained SD300-VGG16 model and evaluate its performance with different thresholds, to then test whether transfer learning improves the overall accuracy.

Experimental design: We assess the pretrained model on 10 videos which showcased all the different sheep activities on three different thresholds: 0.1, 0.4 and 0.7. This allows us to attest to whether sheep were detected at all, even at lower confidence levels. Since the COCO dataset also includes labels for other animals, we included those that could be confounded with sheep like cows and horses, allowing us to check whether the model actually detected animals even though not the correct one.

Results: With a threshold of 0.4 the overall precision was 0.9, meaning that the positive results were usually correct. Nonetheless, the recall resulted in a value of 0.2 when only considering sheep, meaning that the model missed many positive instances. Looking solely at sheep, we get an F1 score of 0.33, while considering other similar animals, we acquired a higher result of 0.49. These results show us that there is indeed a trade-off between precision and recall. When looking at the MSE and MAE we get errors of over 1500, indicating that the model's predictions are deviating from the actual values.

For the other two tested thresholds, we see a pattern. For 0.1, even though the precision drops to 0.43, the recall increases to 0.47, meaning that the model can identify more relevant instances but also an increased amount of false positives. For a threshold of 0.7, we see the opposite, a very high precision of 0.97 but recalls below 0.2. Across all thresholds we get an IOU below 0.005, meaning that the model performed poorly and could not align the predicted bounding box with the ground truth, hence, producing a low accuracy.

By analysing the frames, we can observe that the model performs better when the whole body of the sheep is displayed. However, as seen in Figure 4, the model presents problems when their bodies are partially occluded.

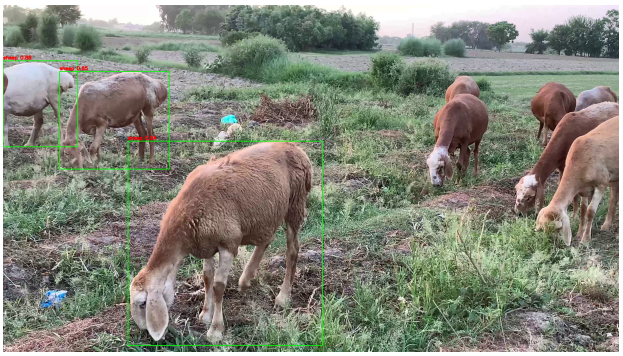

Figure 4. Non-occluded sheep detected by the pretrained model.

Conclusions: Testing the pretrained model onto our dataset gave poor performance results. This makes sense since we are using diverse data with different backgrounds, multiple occluded sheep and behaviours. Consequently, it is necessary to train the model with our videos in order to learn the necessary parameters, which is done via transfer learning. Additionally, the differences between the results obtained in this experiment and the previous one showcase the richness of our chosen dataset. While SSD successfully performs on a simpler dataset with high accuracy, here it clearly shows larger errors when faced with more challenging data, such as in real-world scenarios

5.3. Transfer Learning with SSD300-VGG16 model

Description of the experiment: The goal of the experiment is to analyse any potential improvements for our dataset when using transfer learning. For this, we used the SSD300-VGG16 model pretrained with the COCO dataset. During training, we used SGD and a step learning rate scheduler.

Experimental design: As it was previously explained, for transfer learning the weights of our model were initialized with the pretrained weights of the SSD300-VGG16 model. After that, we trained our model with our dataset, and it was later tested. For this, we used the common 80-10-10 split for training, validation and test sets. The results obtained from experiment 5.2 were expected to be improved since the weights from the pretrained SSD model could be updated to better match our data. As this model was trained on the very large dataset COCO, it has acquired a robust understanding of various features and patterns present in the data.

Results: The obtained results did not match our expecta-

tions, as the errors, both MSE and MAE, were significantly high. Given that, during training, the model showcased really large errors too, this implies that SSD did not correctly capture the underlying patterns in the training data. Consequently, we observe low values for precision, recall, and F1, as seen in Table 3. By monitoring the model parameters during training, we observed a significant increase in loss during the first epoch. Subsequently, the loss decreases gradually and gets stalled around epoch 4. The initial peak suggests that the model fails to trail the training from the beginning. Thus, we may benefit from lowering the starting learning rate or changing the optimizer. The stalling losses (and weights) may indicate some form of vanishing gradient descent, or alternatively, be the result of very low learning rates that compromise the training.

Nonetheless, for certain frames we did observe some promising detections as shown in Figure 5. Even though the sheep that is closer to the camera is not detected, the model does manage to detect some of the occluded and far away sheep. This indicates that even considering the bad overall results, the model exhibits potential by identifying some complex patterns.

Conclusions: It is clear that these results are not optimal, therefore, some further refinement or modifications of the model were needed to improve its performance. Even though transfer learning has proven to be a successful approach, given the limited computational resources, we were only able to use a subset of our dataset. This could have had a direct impact on the performance of the model. The following experiment will have the purpose of tackling this limitation.

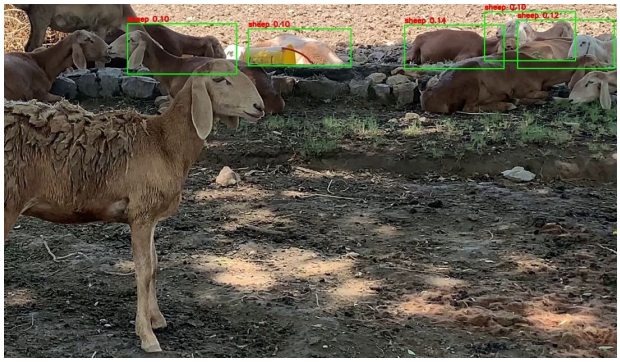

Figure 5. Sheep detected by the model of transfer learning with SSD300-VGG16.

5.4. Transfer Learning, Data Augmentation and Optimisation Techniques with SSD300-VGG16

Description of the experiment: The objective of this experiment is to fine-tune the model using data augmentation and different optimisation techniques, in order to find one that improves the performance. We focus on the methods of data augmentation through cropping, and the Adam optimiser as an alternative to SGD, examining their individual and combined effects when used alongside transfer learning.

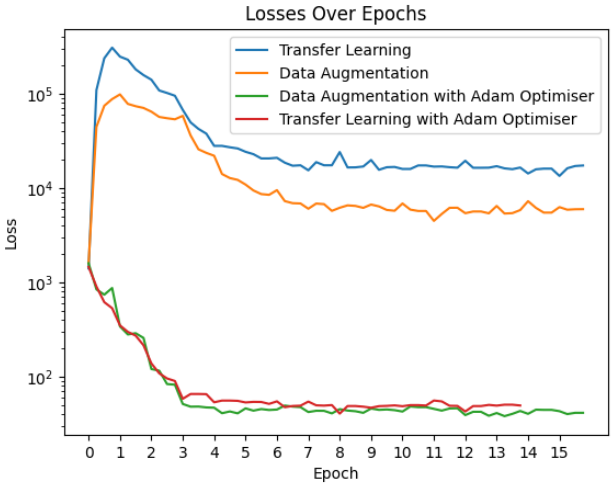

Figure 6. Overall loss on our experimental models: Transfer Learning, Transfer Learning with Data Augmentation, Transfer Learning with Data Augmentation and the Adam Optimizer, and Transfer Learning with the Adam Optimizer.

Experimental design: Due to limited computing resources, the size of our dataset had to be reduced. Therefore, to increase data diversity without having to process more videos and frames we can apply data augmentation transformations such as cropping. This will aid the model to generalise better across different types of input data, to make it more robust to unseen frames. To apply the technique, we took two squared crops vertically centred from every frame used in the training. These were then resized to the standardised dimension of 300x300 pixels. Consequently, the model could abstract more information without the need to load more videos. In addition, given that the original frames were significantly larger than the dimensions used in the model, resizing them directly could lead to loss of detail, disproportionate-sized sheep and distortions. By cropping the images, we addressed these issues by using appropriately sized crops and, since every frame now produced two inputs for the model, it allowed us to reduce by half the training data, using a total of 25 videos.

The second optimising technique involves the Adam optimiser with a starting learning rate of 0.001, which dynamically adjusts the learning rates for each parameter during training. With this, we aim to tackle the initial peak in loss exhibited in the previous experiment. We also hope to achieve faster convergence and better overall performance since Adam incorporates momentum, which can accelerate the learning process.

Results: When using data augmentation, with a threshold of 0.4, we obtained a precision of 2.81e-05 and a recall of 0.0001. Although this is an improvement from doing transfer learning alone, it only achieved one true positive on the test data, while detecting 35623 false positives. On the other hand, when using the Adam optimiser the precision improved up to 0.0026 and the recall to 0.01. Although they are considered poor results, the true positives went up to 89, showing the efficacy of this optimiser.

Threshold	MAE	MSE	Precision	Recall	F_1
0.1	11098.8	1231833614.4	1.90e-02	9.21e-02	3.15e-02
0.4	11213.9	1257515532.1	4.32e-05	2.04e-04	7.13e-05
0.7	11213.7	1257470676.9	4.32e-05	2.04e-04	7.13e-05

Table 3. Experiment 5.3 results

Model	Threshold	MAE	MSE	Precision	Recall	F_1	TP
TL	0.4	11213.9	1257515532.1	4.32e-05	2.04e-04	7.13e-05	4
TL+DA	0.4	4465.4	199397971.6	2.81e-05	1.11e-04	4.48e-05	1
TL+AO	0.4	4354.3	189599284.9	2.57e-03	9.85e-03	4.07e-03	89
TL+DA+AO	0.4	3767.3	141925492.9	2.46e-03	7.87e-03	3.76e-03	71

Table 4. Experiment 5.4 results on the use of Data Augmentation and the Adam Optimiser.

Since both techniques improved the baseline model, we combined them to assess whether it would improve the performance any further. However, from our experiments, we discovered that the combination of data augmentation and Adam optimizer slightly underperformed the Adam optimiser alone, since it achieved a total of 71 true positives and a precision of 0.0025.

Figure 6 compares the loss on each one of these experiments. We can clearly observe how models that used the Adam optimizer have overcome the initial peak in loss exhibited in the previous experiment, as opposed to those still implementing SGD. We also see how data augmentation slightly improves performance compared to our baseline transfer learning model. However, these results also show the persistent problem of stalling losses, which makes the training and successful convergence of our model difficult. The models' performance scores are shown in Table 4.

Conclusions: These experiments showed that optimising techniques such as data augmentation and using an Adam optimiser did indeed improve the performance of the model. However, although the results were slightly enhanced, the model still performed poorly overall. To overcome the problem of stalling weights, further experiments need to be carried out, plausibly considering different learning schedulers or deepening into an analysis of gradient flow.

6. Analysis

Our dataset did not only include annotations about the specific activity performed by the sheep (either walking, sitting, standing, grazing or running) but it was also labelled in regards to the video showing occlusion or not. Previous relevant research has not analysed the impact of sheep activities on the model's performance and therefore we deemed it necessary to do further analysis on this topic. On the other hand, occlusion is one of the greatest challenges within the object detection field, hence the relevance of studying its impact on the model's reliability.

Over the multiple experiments performed, the activity which obtained better results was grazing and the worst was running. This is rather interesting and implies that the fact that the sheep is not static will hugely impact the effectiveness of the model. Furthermore, as it was already expected, the videos with occlusion had overall worse metrics. This confirms the well-known challenge in object detection of occlusion.

7. Conclusions

The diversity of background, lighting and occlusion in the videos of our dataset presented multiple challenges for the task of sheep detection. Even though transfer learning did not exhibit optimal results, there was an observable improvement with the application of optimisation techniques: data augmentation and the Adam optimiser.

Throughout the analysis of our experiments, we have identified plausible causes for the problems encountered, including a possible Vanishing Gradient problem and alternatively the need to further fine-tune the learning rate or use of a different scheduler. On the other hand, we have presented results that showcase early stages of improvement in the detection of sheep in our new dataset. Undoubtedly, the outcomes of our experiments reflect the richness of our newly presented data and the challenges present within, ranging from occlusion, to background diversity and motion distortion.

This work provides a range of preliminary experiments to serve as a baseline for the extensive research still pending to be done on this task. It is clear that further experiments still need to be carried out, with enough resources to tackle the entire extent of the data. As a starter, we propose as the next step the analysis of different learning schedulers to overcome the problem of stalling losses, and/or seeing the effects of larger training sets on the already presented experiments. Additionally, the modification of the internal structure of the model layers, such as implementing residual connections, different activations, etc. could aid with vanishing gradients.

References

Bock, Sebastian, Goppold, Josef, and Weiß, Martin. An improvement of the convergence proof of the adam-optimizer. *arXiv preprint arXiv:1804.10587*, 2018.

Hardt, Moritz, Recht, Ben, and Singer, Yoram. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

Kelly, Nathan A., Khan, Bilal M., Ayub, Muhammad Y., Hussain, Abir J., Dajani, Khalil, Hou, Yunfei, and Khan, Wasiq. Video dataset of sheep activity for animal behavioral analysis via deep learning. *Data in Brief*, 52: 110027, February 2024. ISSN 2352-3409. doi: 10.1016/j.dib.2024.110027. URL https://www.sciencedirect.com/science/article/pii/S2352340924000015.

Khan, Bilal and Kelly, Nathan. Video Dataset of Sheep Activity (Grazing, Running, Sitting). 1, October 2023. doi: 10.17632/h5ppwx6fn4.1. URL https://data.mendeley.com/datasets/h5ppwx6fn4/1.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, Wei, Anguelov, Dragomir, Erhan, Dumitru, Szegedy, Christian, Reed, Scott, Fu, Cheng-Yang, and Berg, Alexander C. SSD: Single Shot MultiBox Detector. volume 9905, pp. 21–37. 2016. doi: 10.1007/978-3-319-46448-0_2. URL http://arxiv.org/abs/1512.02325. arXiv:1512.02325 [cs].

Morris, Steven. "it is devastating": Uk farmers despair as sheep thefts soar, Feb 2020. URL https://www.theguardian.com/environment/2020/feb/02/it-is-devastating-uk-farmers-despair-as-sheep-thefts-soar.

Ninja, Dataset. Visualization tools for sheep detection dataset. https://datasetninja.com/sheep-detection, mar 2024. URL https://datasetninja.com/sheep-detection. visited on 2024-03-25.

Qassim, Hussam, Verma, Abhishek, and Feinzimer, David. Compressed residual-vgg16 cnn model for big data places image recognition. In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pp. 169–175. IEEE, 2018.

Sarwar, Farah, Griffin, Anthony, Periasamy, Priyadharsini, Portas, Kurt, and Law, Jim. Detecting and Counting Sheep with a Convolutional Neural Network. pp. 1–6, November 2018. doi: 10.1109/AVSS.2018.8639306.

Sarwar, Farah, Griffin, Anthony, Rehman, Saeed Ur, and Pasang, Timotius. Detecting sheep in UAV images. *Computers and Electronics in Agriculture*, 187:106219, August 2021. ISSN 0168-1699. doi: 10.1016/j.compag.2021.106219. URL https://www.sciencedirect.com/science/article/pii/S0168169921002362.

Wang, Liang, Hu, Bo, Hou, Yuecheng, and Wu, Huijuan. Lightweight Sheep Head Detection and Dynamic Counting Method Based on Neural Network. *Animals*, 13(22):3459, January 2023. ISSN 2076-2615. doi: 10.3390/ani13223459. URL https://www.mdpi.com/2076-2615/13/22/3459.

Wollan, Malia. How to Count Sheep. *The New York Times*, September 2017. ISSN 0362-4331. URL https://www.nytimes.com/2017/09/08/magazine/how-to-count-sheep.html.

Xu, Jianming, Liu, Weichun, Qin, Yang, and Xu, Guangrong. Sheep Counting Method Based on Multiscale Module Deep Neural Network. *IEEE Access*, 10:128293–128303, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3221542. URL https://ieeexplore.ieee.org/abstract/document/9945968. Conference Name: IEEE Access.

Yu, Chen, Zhou, Jiehan, Song, Xianhua, and Lu, Zeguang. *Green, Pervasive, and Cloud Computing: 17th International Conference, GPC 2022, Chengdu, China, December 2–4, 2022, Proceedings*. Springer Nature, January 2023. ISBN 9783031261183. Google-Books-ID: LRKrEAAAQBAJ.