Detecting and Counting Sheep with a Convolutional Neural Network

Conference Paper · November 2018

DOI: 10.1109/AVSS.2018.8639306

5 authors, including:

Farah Sarwar
University of Management and Technology (Pakistan)
8 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Anthony Griffin
Auckland University of Technology
43 PUBLICATIONS **541** CITATIONS
SEE PROFILE

Detecting and Counting Sheep with a Convolutional Neural Network

Farah Sarwar^{1,2}, Anthony Griffin^{1,2}, Priyadharsini Periasamy^{2}, Kurt Portas^{3} and Jim Law^{3}

^{1} High Performance Computing Research Lab, ^{2} Electrical and Electronic Engineering Department
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology, New Zealand
farah.sarwar@aut.ac.nz, anthony.griffin@aut.ac.nz, peripriya86@gmail.com

^{3} Palliser Ridge Limited, Pirinoa, New Zealand
kurt_tp@hotmail.com, jimalaw@yahoo.com

Abstract

Counting livestock is generally done only during major events, such as drenching, shearing or loading, and thus farmers get stock numbers sporadically throughout the year. More accurate and timely stock information would enable farmers to manage their herds better. Additionally, prompt response to any stock in distress is extremely valuable, both in terms of animal welfare and the avoidance of financial loss. In this regard, the evolution of deep learning algorithms and Unmanned Aerial Vehicles (UAVs) is forging a new research area for remote monitoring and counting of different animal species under various climatic conditions. In this paper, we focus on detecting and counting sheep in a paddock from UAV video. Sheep are counted using a model based on Region-based Convolutional Neural Networks and the results are then compared with other techniques to evaluate their performance.

1. Introduction

Many industries want more data about the operation of their business, and livestock farmers are no different. With the advent of precision agriculture [9], many farmers have more data than ever before, such as ground temperatures, soil moisture, pasture covers, and individual stock identification to monitor growth rates and welfare. However, precise and timely information about the numbers and location of their animals is hard to come by, particularly on larger farms. Indeed, counting livestock still remains a manual task that is labour-intensive and prone to minor—but significant—errors. It can also be disruptive to the animals, as they must be run through a drafting race or a narrow choke point. Currently, many farmers only count their animals once a month or so [14], and some farmers only count their stock when they are being loaded on or off a truck, as they are arriving or leaving the farm. Thus, many farmers are interested in monitoring their livestock using a robust automated system. Monitoring the distribution and population of animal species with the passage of time is also a key ingredient to successful nature conservation [21]. Animal observation can be used for multiple tasks such as monitoring animal growth, animal distress, conducting surveys, loading and unloading from holding pens, etc. This variety of applications has motivated researches to develop some sophisticated—yet faster—ways to achieve this target, but this is still an emerging field as little work has been done so far. Relevant to this research area, many challenges such as diversity of background, species-specific characteristics, spatial clustering of animals [18] arise. Researchers use different statistical and biological methods [6, 10] to cope with these challenges. Some have adopted different approaches and tried to solve similar tasks using techniques like a template matching algorithm [22], AdaBoost classifier [1], power spectral based techniques [13], Deformable Part-based Model (DPM), Support Vector Machine (SVM) [21] and Convolutional Neural Networks (CNNs) [2] for automatic processing and showed some good results. However, all of them have tested their techniques on images where the animals are few in number and occupy a majority of the image with high resolution. We wish to deal with images having hundreds of small animals per image.

In this work we focus on counting sheep from Unmanned Aerial Vehicle (UAV) video. Although this is easier than counting cattle or other animals—due to the relatively uniform colour of sheep—it is not a trivial task, particularly when dealing with hundreds or even thousands of sheep. The goal of this work is to provide farmers with extremely accurate information about how many sheep are in each of their paddocks, using a commonly-available UAV to count the sheep in situ. This involves no disruption to the animals, the UAV is so high that they rarely notice it.

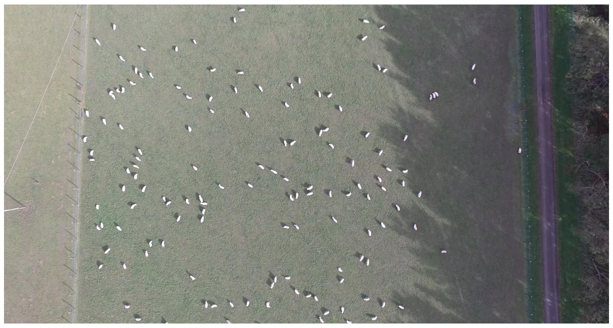

Figure 1. Example test image, taken at 80 m from the ground containing 139 sheep.

	Training		Testing	
Data Set	No. of Images	Total Sheep	No. of Images	Total Sheep
Sunny	2	267	4	582
Overcast	2	319	4	520
Mixed	4	586	8	1102

Table 1. Details of data sets used for both methods. All images are $(2048 \times 1080 \times 3)$ pixels in size.

Counting different objects in images was initially performed using hand-crafted representations that need feature extraction for respective items [19] and then shifted to deep learning algorithms to increase the speed and accuracy. Currently, the researchers are focusing on solving object detection using CNN [5, 11, 12, 17]. In the last few years, many researchers have provided the modified structures of CNN based models and delivered outstanding results in classification tasks such as Ciresan [3], which demonstrated very good performance on the NORB and CIFAR-10 datasets. Krizhevsky [7] beat the record of classification on ImageNet 2012. Since then extended versions of CNNs are being proposed such as, Region-based CNN (R-CNN) [5], Fast R-CNN [4], Faster R-CNN [16], You Only Look Once (YOLO) [15] and Single Shot MultiBox Detector (SSD) [8]. Amongst all these techniques, YOLO is the fastest but makes more localization errors as compared to Faster R-CNN and is not suitable for detection of tiny objects [15]. Faster R-CNN provides best results so far but is prone to false positives in the background.

We now discuss our methodologies in Section 2, and Section 3 will discuss the experimental results and compare two different methods to detect and count sheep in aerial images. Conclusions are presented in Section 4.

2. Methodology

In this section we present two different methods for sheep detection in a frame. Method A is based on a basic architecture of R-CNN, while Method B uses hand crafted techniques. Results of both these techniques are compared and discussed in later sections.

Data set	No. of Images	Total Sheep
Sunny	146	884
Overcast	169	811
Mixed	318	1706

Table 2. Details of augmented training data sets for method A. All images are $(250 \times 250 \times 3)$ pixels in size.

Our main task is to detect and count the sheep using images of a paddock, which were taken from a UAV recorded video in Palliser Ridge Ltd, a sheep and beef farm in the Pirinoa region of New Zealand.

There are commonly hundreds of sheep in a 2048×1080 RGB video frame, and each sheep is roughly 10×20 pixels. The video was recorded at a height of 80 m above the ground. Both methods use the data sets shown in Table 1, for training and testing, where there a total of 4 images used for training and 8 for testing. There is no overlap between the images used for training and testing. Figure 1 shows an example of one of the training images.

2.1. Method A: R-CNN

Method A is mainly composed of 2 parts: designing training data, and applying R-CNN for detection.

2.1.1 Designing Training Data

The training images were very similar to each other, so in order to augment the training data for Method A we cropped out sub-images of 250×250 pixels and rotated them to different angles. From the 4 full-size images, we designed almost 400 small images to complete the training data set. Among these 400 images, 82 images contained no object of interest and have only background. We then designed three different training data sets using rest of the 318 images; sunny, overcast and a mix of overcast and sunny images. Table 2 shows the number of images and sheep per image in the augmented training data sets for Method A.

Each training image is of $250 \times 250 \times 3$ size and all the sheep are labelled manually in them. This was quite a lengthy process as the number of sheep per training image varies in the range of [0, 18]. We used RGB colour images to keep this algorithm applicable for different animal species for our later research. There are a total of 1706 objects (sheep) or Region of Interests (ROIs) in our complete training data set. Those objects of interest which were not completely in an image or were at the boundaries were not labelled. If we had kept all such objects then it would only increase the false positive detection on the background and thus degrade our results. Each ground truth bounding box has four components: (x, y, w, h), where x and y represents

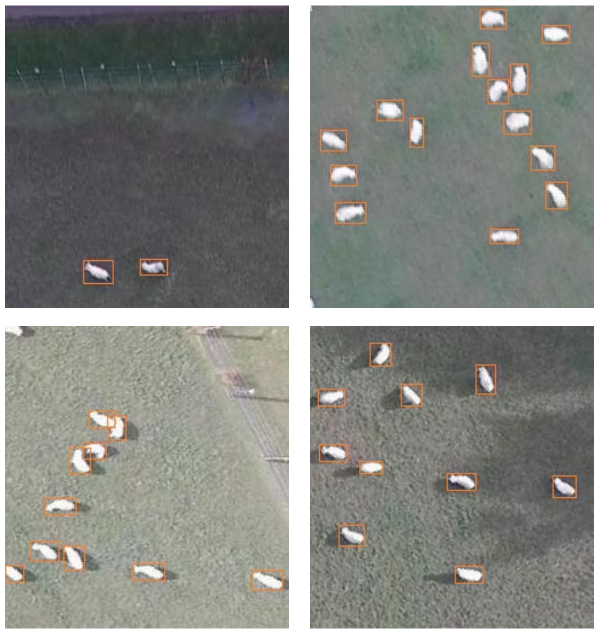

Figure 2. Annotated training images.

the starting coordinates of the respective ROI and w and h gives its dimension in terms of width and height respectively. Example of overcast, sunny and partly-sunny images with labelled sheep are shown in Figure 2.

2.1.2 Applying R-CNN for Detection

The R-CNN based object detection algorithm consists of three different modules [5]:

1. Region proposals
2. Convolutional Neural Network
3. Class-specific linear Support Vector Machines (SVMs)

Girshick *et al*. [5] used the selective search algorithm [20] for defining region proposals. The input images of training data set are warped to $277 \times 277 \times 3$ pixel size internally before the computation of region proposals. Afterwards, features are extracted from each region proposal using CNN. The output of this layer is then fed to an SVM classifier as well as a simple linear bounding box regressor to get a confidence value of classification for each bounding box. The centroids of each sheep are given by the centres of the bounding boxes.

The network used by Girshick [5] has 4 convolutional layers and 2 fully connected (FC) layers. We have used different architectures to check the impact of different combinations of convolutional and FC layers. The main algorithm for processing the input data remains the same but is applied to different network architectures. Figure 3 shows the starting network structure. We then systematically eliminated

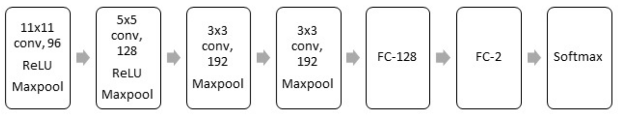

Figure 3. Main architecture of R-CNN network.

convolutional layers (from right to left) to analyze the impact of smaller architecture on our results. And surprisingly the results were not affected to a very large extent. A decrease in precision of 5% and increase in recall of 6% was observed when going down to a network with 1 initial convolutional layer and 1 fully connected layer of 2 neurons followed by soft max layer.

2.2. Method B: Expert System

Method B is a hand-crafted technique that takes advantage of the fact that in all the images, the sheep are quite uniformly white, and generally the brightest objects in an image. We used blob analysis on a brightness thresholded greyscale image. Our detection algorithm generates $\hat{\mathbf{c}}_i$, the estimated centroids of the sheep in the i-th image, as follows:

1. Generate the greyscale image G_i based on the RGB image from the video.
2. Generate a binary image B_i whose pixels are 1 if the corresponding pixel in G_i is above a minimum brightness threshold β and 0 otherwise.
3. Find and label all the 8-connected blobs in B_i.
4. Remove the blobs with an area less than a minimum area threshold α.
5. Record the centroids of the remaining blobs as $\hat{\mathbf{c}}_i$.
6. Calculate the mean of the blobs' areas as $\bar{A}$.
7. Record the blobs whose area is greater than $\gamma\bar{A}$ as two-sheep blobs, by duplicating these centroids.

Note that β ranges from 0 to 1, α is an integer number of pixels, γ is a number a little less than 2, and $\hat{\mathbf{c}}_i$ is an $N_i \times 2$ matrix, whose j-th row is denoted by $\hat{\mathbf{c}}_i^{(j)}$ and contains the x and y pixel locations of the centroid of the j-th sheep. Obviously, N_i is the number of sheep detected in the i-th image.

Although sheep tend to spread out when left alone in a paddock, often there will be a group of two or more sheep that are quite close together. Computing the centroids using only first five steps of the algorithm can fail to identify multiple sheep for such groups. So, the last two steps of the algorithm were added to improve the counting accuracy.

In order to test the accuracy of this algorithm we manually labelled the centroids of sheep in the training image. Let these ground truth centroids be denoted by $\mathbf{c}_i$. We then wish to compare $\mathbf{c}_i$ and $\hat{\mathbf{c}}_i$ to get a measure of the error, however $\mathbf{c}_i$ and $\hat{\mathbf{c}}_i$ may contain differing numbers of rows and their rows are likely to be in different orders. Furthermore, the most similar rows of $\mathbf{c}_i$ and $\hat{\mathbf{c}}_i$ are likely to not be exactly equal due to differences in human and machine processing.

For each row of $\mathbf{c}_i$ we calculate the following:

$$d = \min_k \left\| \hat{\mathbf{c}}_i^{(j)} - \mathbf{c}_i^{(k)} \right\|_2 \tag{1}$$

$$k' = \arg\min_k \left\| \hat{\mathbf{c}}_i^{(j)} - \mathbf{c}_i^{(k)} \right\|_2 \tag{2}$$

And we say that $\hat{\mathbf{c}}_i^{(j)}$ matches $\mathbf{c}_i^{(k')}$ if $d < \delta$, where δ is an error distance threshold in pixels, set at 10 pixels. We must ensure that we don't match $\mathbf{c}_i^{(k')}$ to another row of $\hat{\mathbf{c}}_i$, so we remove the k'-th row of $\mathbf{c}_i$, before proceeding to the next row of $\hat{\mathbf{c}}_i$. Our error metric is the number of unmatched rows of $\mathbf{c}_i$ plus the number of unmatched rows of $\hat{\mathbf{c}}_i$.

In order to test the sensitivity of this detection method to the two parameters α and β, we varied them both over a large range, and calculated the error described above. Figure 4 shows the error for the example frame in Figure 1, which has bright, white sheep, and varying illumination. It is clear that there are quite wide ranges of α and β that perfectly detect the sheep.

3. Results

In order measure the performance of the two detection methods, we followed the procedure in Section 2.2, to compute the overlap between the true centroids and the estimated centroids of the sheep. We calculated the precision and recall, defined as:

$$\text{Precision} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}} \tag{3}$$

$$\text{Recall} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}} \tag{4}$$

where $N_{\text{TP}}, N_{\text{FP}}, N_{\text{FN}}$ are the number of true positive, false positive and false negative sheep detections, respectively.

Obviously, the ideal is a system that has a precision and recall of 100%. However, in a counting context, recall is more important as it measures how many of the sheep were detected, whereas precision measures how confident you can be that a detected sheep is actually a sheep.

3.1. Method A

Tuning most of the hyper parameters for successful implementation of R-CNN deals with pragmatic evidence and testing. We experimented with different settings to determine the combination of parameters to provide the best results. While testing with different training options we observed that a mini batch size beyond a certain range did not

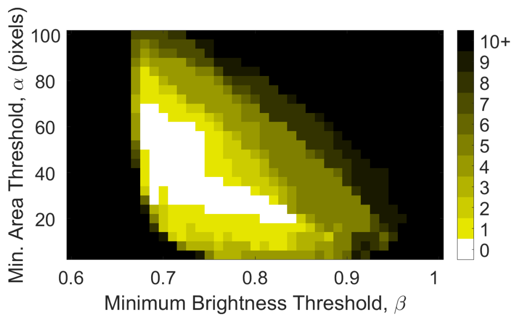

Figure 4. Sensitivity matrix for the frame in Figure 1 for Method B, the colours represent the number of detection errors, ranging from white being no errors, to black being 10 or more errors.

give good results for our data set. The algorithm took too long to find a minimum error for batch size more than 32 and needed too many epochs below the value of 10. Similarly, different variations in results were observed when we tried a negative overlap region between 0.1 and 0.5. Girshick *et al.* [5] used the value of 0.3 in their experiments as it best suited their data set, but it did not work for our case. When we decreased the value to 0.2, it improved the results with 7% recall, which was a huge difference. Another important parameter is the initial learning rate, which also depends on the training data set and the type of object under observation. Our network works well with this value being around 10^{-5} but above this value it gets stuck in a local minimum and below this value it just oscillates and takes too long to settle down somewhere. So, we used a mini batch size of 15, an initial learning rate of 10^{-5} and negative overlap region of 0.2 while training on our data sets.

Along with adjusting the options for training, we also checked results using different architectures for the network. These architectures are as follows:

1. 96 filters with different kernel sizes of 5×5, 11×11, 21×21, 25×25 and 29×29 in a network of 1 convolution layer followed by 1 FC and softmax layer.
2. R-CNN network with 2, 3, 4, 5 and 6 network layers using a 6-layer network structure as shown in Figure 3.
3. A 6-layer R-CNN network using different kernel sizes in each convolutional layer.

The graphs of precision and recall using previously discussed hyper parameters and above mentioned architectures are shown in Figure 5. It can be seen that the use of larger kernel size of around 21×21 and 25×25 give good results when compared to other options. These filters are tested only in the smallest network as different filter combinations in larger networks did not give significant variations, as can be observed from the bottom right graph in the same figure. Smaller filters usually capture minute details of objects and

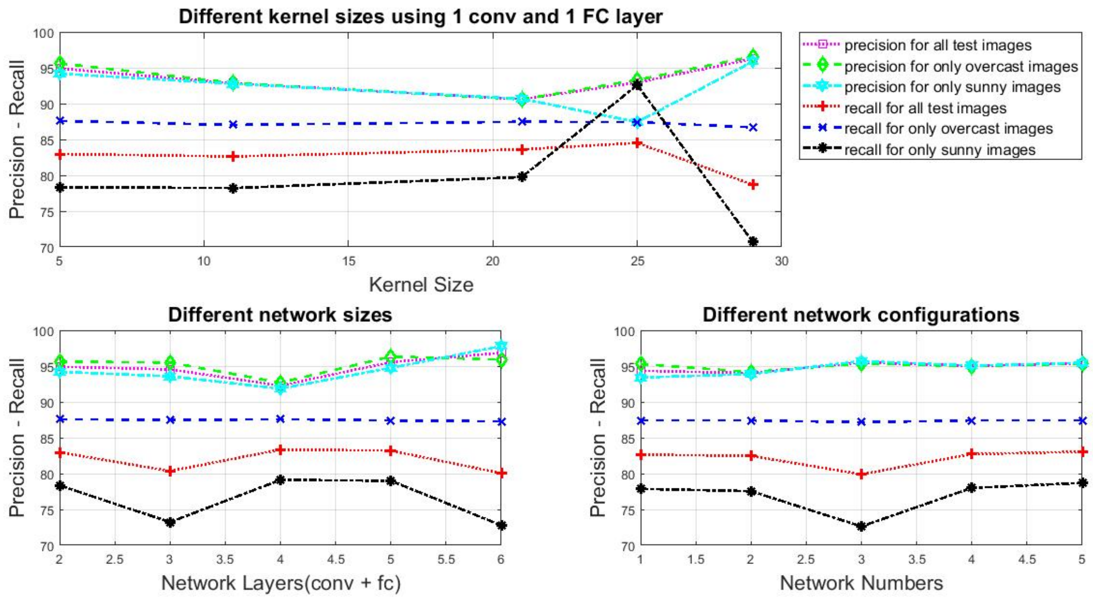

Figure 5. Precision and Recall curves using different settings. All figures show three curves of precision and three curves of recall for three test data sets. The top left figure was obtained using a 2-layer R-CNN network (1 conv + 1 FC + softmax) with the shown kernel sizes for the convolutional layer. The bottom left figure shows the respective curves for 2-, 3-, 4-, 5- and 6-layer R-CNN networks. The bottom right shows results with a 6-layer R-CNN network using different kernel sizes in each network.

our current object is an oval shaped white blob. Thus, having only one oval shape white blob removes the necessity for larger network structures.

Different training data sets also contribute towards the accuracy of the results. It was observed that although the training set with only sunny images has half the number of images when compared to the full data set, it was able to train the network in almost the same way. The results were quite poor while using training data set of only overcast images. The details of test images were presented in Table 1, which were used for testing all discussed networks. Results presented in Figure 5 were computed using the training data set only.

It can be seen that best recall is observed for cloudy test images as there is less illumination variations in them. And highest recall for both types of test data sets is observed using network of 2 layers and 96 (21×21) filters in the convolutional layer. One such resultant test image is shown in Figure 6.

3.2. Method B

Once the thresholds α, β and γ had been tuned on the 4 training images, Method B was applied to the 8 test images. The values chosen were

$$\alpha = 35, \qquad \beta = 0.75, \qquad \gamma = 1.6 \tag{5}$$

The precision and recall for Method B are shown in Table 3. This method performs perfectly on the training images, and only shows only slight degradation on the test images.

Data set	Precision	Recall
Training	100.0%	100.0%
Testing	95.6%	99.5%

Table 3. Precision and Recall for Method B

3.3. Discussion

While it is clear that the Method B outperforms Method A, the CNN method does show some promise. Indeed there is perhaps a way to combine the two methods to achieve even better detection performance. When Method B fails, it is due to an increase in false negatives, particularly around background objects such as trees. Method A tends to have very low false negatives, so perhaps this information can be used to help Method B improve its discrimination.

4. Conclusion

In this work we have taken an important step forward in the previously unexplored area of computer vision techniques to detect and count sheep from UAV video. We have used two totally different approaches to handle the task, which both work well, although it can be seen from results that more work is required for deep learning algorithms to perform effective object detection, especially if the objects are of very small size as compared to the background. The

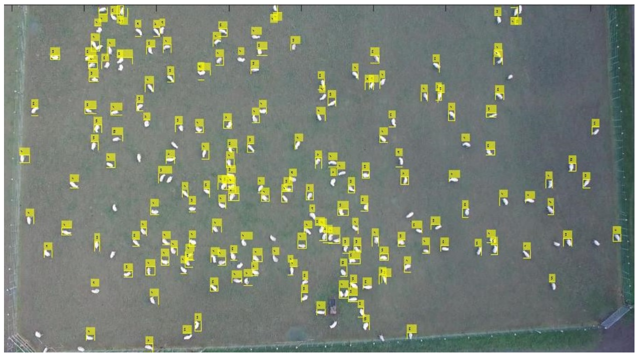
Figure 6. 189 detected sheep with 9 false positive results in an overcast test image having true count of 199, for method A.

hand-crafted method proposed here shows great promise for sheep detection and counting.

Nonetheless, there are many areas still open for further work, such as speeding up the algorithms, detecting the fence lines, improving the robustness of the algorithms to parameter selection, and the use of tracking to cope with small mistakes in object detection. CNN-based techniques may be of use in background discrimination, as well as detecting less uniform livestock such as cattle, which can be a variety of colours, or even multi-coloured. We hope that this work will inspire other researchers to tackle some of these challenges.

References

[1] T. Burghardt and J. Calic. Real-time face detection and tracking of animals. In *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, pages 27–32. IEEE, 2006.

[2] P. Chamoso, W. Raveane, V. Parra, and A. González. UAVs applied to the counting and monitoring of animals. In *Ambient Intelligence-Software and Applications*, pages 71–80. Springer, 2014.

[3] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 3642–3649. IEEE, 2012.

[4] R. Girshick. Fast R-CNN. *arXiv preprint arXiv:1504.08083*, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[6] J. Ingram and J. Anderson. Tropical soil biology and fertility. *A handbook of methods. 2nd ed. CAB Int., Wallingford, UK*, 1993.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[9] A. McBratney, B. Whelan, T. Ancev, and J. Bouma. Future directions of precision agriculture. *Precision agriculture*, 6(1):7–23, 2005.

[10] J. McKinlay, C. Southwell, and R. Trebilco. Integrating count effort by seasonally correcting animal population estimates (icescape): A method for estimating abundance and its uncertainty from count data using Adélie penguins as a case study. *CCAMLR Science*, 17:213–227, 2010.

[11] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1717–1724. IEEE, 2014.

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? – Weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015.

[13] M. Parikh, M. Patel, and D. Bhatt. Animal detection using template matching algorithm. *International Journal of Research in Modern Engineering and Emerging Technology*, 1(3):26–32, 2013.

[14] K. Portas. personal communication, December 2015.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[18] G. Sileshi. The excess-zero problem in soil animal count data and choice of appropriate models for statistical inference. *Pedobiologia*, 52(1):1–17, 2008.

[19] V. A. Sindagi and V. M. Patel. A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16, 2017.

[20] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[21] J. C. van Gemert, C. R. Verschoor, P. Mettes, K. Epema, L. P. Koh, and S. Wich. Nature conservation drones for automatic localization and counting of animals. In *Workshop at the European Conference on Computer Vision*, pages 255–270. Springer, 2014.

[22] F. A. Wichmann, J. Drewes, P. Rosas, and K. R. Gegenfurtner. Animal detection in natural scenes: critical features revisited. *Journal of Vision*, 10(4):1–27, 2010.