A Comparison of Convolutional Neural Network Architectures for Automated Detection and Identification of Waterfowl in Complex Environments

Article *in* GI_Forum · January 2021

DOI: 10.1553/giscience2021_02_s152

4 authors:

Mohammad Sa'Doun
Fachhochschule Kärnten
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Christopher D. Lippitt
University of New Mexico
66 PUBLICATIONS **1,492** CITATIONS
SEE PROFILE

Gernot Paulus
Fachhochschule Kärnten
50 PUBLICATIONS **417** CITATIONS
SEE PROFILE

Karl-Heinrich Anders
Fachhochschule Kärnten
45 PUBLICATIONS **729** CITATIONS
SEE PROFILE

GI_Forum 2021, Issue 2
Page: 152 - 166
Full Paper
Corresponding Author:
mohammad.sadoun.08@gmail.com
DOI: 10.1553/giscience2021_02_s152

A Comparison of Convolutional Neural Network Architectures for Automated Detection and Identification of Waterfowl in Complex Environments

Mohammad Mustafa Sa'doun[1], Christopher D. Lippitt[2], Gernot Paulus[1] and Karl-Heinrich Anders[1]

[1]Carinthia University of Applied Sciences, Villach, Austria
[2]University of New Mexico, Albuquerque, USA

Abstract

Waterfowl monitoring is an important task for understanding waterfowl distribution and habitats. Surveying approaches using hyper-spatial airborne imagery, collected by small unoccupied aerial systems (sUAS), hold potential to overcome the limitations of traditional methods while improving count efficiency and reliability. Difficulties obtaining waterfowl counts, particularly in complex image scenes, from the high quantity of imagery required hinders deployment of large-scale surveys. In this paper, we test Convolutional Neural Networks (CNNs) to understand their potential and how they behave across different versions of our waterfowl dataset. Three CNN architectures (YOLO, Retinanet and Faster R-CNN) were trained on 3 hierarchical levels: waterfowl detection (True / False), waterfowl type (3 classes), and waterfowl species (8 classes). The architectures generally performed well, and results indicate that automated waterfowl detection in complex environments, and therefore enumeration, is feasible using current technology. Waterfowl identification in complex environments was not successful using the available training data, but we propose steps that might enhance the results.

Keywords:

waterfowl surveying, YOLO, Retinanet, Faster R-CNN, sUAS, deep learning

1 Introduction

Waterfowl population recognition and classification have traditionally been undertaken by a combination of ground-based and manned aircraft surveys. Manned aircraft enable the surveying of large areas, but they are expensive and can cause stress to wildlife (Wilson et al., 1991). Small unoccupied aerial systems (sUAS) have been used successfully to survey a variety of bird species worldwide, with much lower costs and risks (Linchant et al., 2015). One factor hindering the adoption of surveys using sUAS is the work required to manually identify targets in the imagery compared with counts in the field (Linchant et al., 2015). Numerous automated techniques have been used for waterfowl recognition, with accuracy comparable to manual image counts, including spectral thresholding (Laliberte & Ripple, 2003), supervised classification (Grenzdörffer, 2013), and template matching (Abd-Elrahman et al., 2005). However, these methods are limited in that they require animals to be highly distinct spectrally from their surroundings. This hinders applications in heterogeneous environments for the study of species with cryptic colouration, or with image sets of varying brightness due to camera performance or weather conditions (Linchant et al., 2015; Chabot & Francis 2016). Some machine learning (ML) approaches, such as convolutional neural networks (CNNs), have the potential to enable efficient detection and classification in complex scenes (Chen et al., 2012). Compared to other ML techniques such as the use of support vector machines and Key Nearest Neighbour, CNNs produce high detection and classification accuracies owing to their non-linearity, ability to increase model complexity (adding convolutional layers for deeper feature extraction), and implicit segmentation capabilities (Ghorbanzadeh et al., 2019; Wang & Raj, 2017). The latest CNNs can be used in many automated processes in various application domains, for example in crowd counting, object detection, face-attribute recognition, and geo-localization (Howard et al., 2017; Girshick et al., 2014; van Gemert et al., 2014; Chen et al., 2012).

CNNs were applied to three hierarchical levels of increasing difficulty, namely waterfowl detection, identification of waterfowl type, and identification of waterfowl species. The selected CNN architectures were trained and validated using a set of labelled sUAS-acquired images of waterfowl to enable comparisons of the architectures' abilities to recognize and classify waterfowl species. The primary aim of this empirical research project was to identify the CNN architecture that can produce the most accurate classifications and counts (relative to the ground truth) as part of an effort to develop a prototype sUAS-based waterfowl survey programme for the United States Fish and Wildlife Service (USFWS).

2 Deep learning and CNNs

Deep Learning (DL) is a sub-category of ML that mimics how the human brain works (Krogh, 2008). It uses Artificial Neural Networks (ANNs) consisting of many layers, each containing neurons (or nodes) connected to form a web-like structure. CNNs are a special type of ANNs, the main difference being that in a CNN a deeper neuron layer is connected only to a subset of neurons in the previous layer (Albawi et al., 2017). Figure 1 shows the basic architecture of a CNN. Many frameworks are available for DL, of which Google's TensorFlow (Abadi et al., 2016) is the latest. The Tensorflow API, which is publicly available, can be used to distribute load between multiple nodes (CPUs or GPUs). Keras is a fast-growing deep learning framework. This open-source library written in Python can run on top of TensorFlow or Theano. Theano is an open-source Python library for numerical computations; it simplifies the process of writing DL models. Caffe, developed by the Berkeley Vision and Learning Centre, has many worked examples of deep learning, written in Python. Giving more importance to GPUs is the Torch framework, which has an underlying C/CUDA implementation and is widely used for DL, as is MATLAB's *matconvnet*.

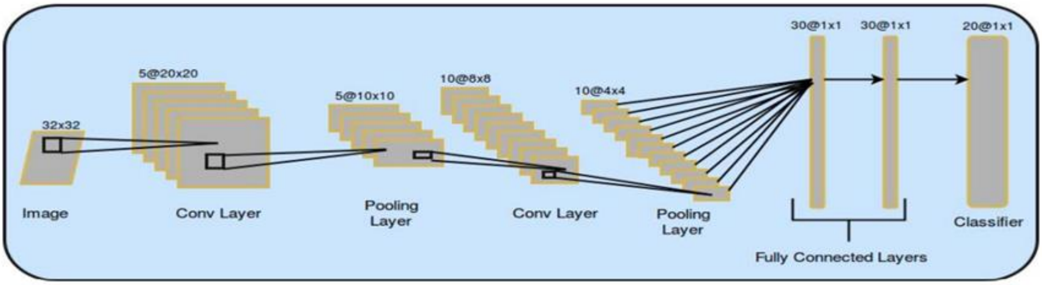

Figure 1: Basic CNN architecture (Aloysius and Geetha, 2017)

Various CNN architectures, such as CifarNet, MobilNets, AlexNet, GoogLeNet or YOLO (Howard et al., 2017; Zha et al., 2015), have accomplished significantly improved performance in many application domains, including object labelling and classification, event detection for safety systems, obstacle avoidance in autonomous driving, and identity checking (Zha et al., 2015). Each architecture differs in the number of intermediary layers, number and size of kernels used, error calculation methods, and activation functions. A feature common to all CNNs, however, is the need for robust training sets that contain both the desired input and the desired output (questions and right answers). The CNN then performs error calculations for its predictions, and runs through further iterations in order to improve predictions.

2.1 Related work using CNNs for bird detection

In a recent study carried out in Korea (Hong et al., 2019), five different CNN architectures were employed for bird detection, namely Faster R-CNN, R-FCN, SSD, Retinanet and YOLO, and were evaluated by comparing their speed and accuracy. The accuracy of the detection was measured using the Intersection over Union (IoU), defined as the ratio of intersection between the predicted box and the ground truth box. A threshold of 0.3 and 0.5 of IoU to determine the acceptability of the detection and the CNNs' performance was measured for both thresholds. The training data comprised 25,864 sUAS images that include 137,486 birds. Although Hong et al. (2019) stated that a 0.3 IoU threshold was sufficient, a threshold of IoU of 0.5 was used in our study. This was because a relatively small number of training samples (not hundreds or tens of thousands) were fed into the CNNs, and a 0.3 IoU led to the CNN twice as many non-waterfowl labels relative as the 0.5 IoU threshold.

2.2 Choice of CNN

A Weighted Linear Combination (WLC) operation was applied to rank the CNNs used by Hong et al. (2019); the three best, based on accuracy defined as an IoU score of 0.3, were selected. It should be noted that not all the CNNs in Table1 were taken into consideration. R-FCN with Resnet 101 was not selected, as this model performed in a manner very similar to Faster R-CNN, with slightly less accuracy but better time. Another reason to suppress R-FCN was that it contains the same feature extractor as Faster R-CNN (Resnet 101) as the core model for detection. Only the IoU of 0.3 was considered because it was sufficient to detect all bird objects. YOLOv2 was also suppressed, because YOLOv3 was developed to obtain better accuracy, which is the prime focus in this study.

First, normalization of the speed and accuracy for each option was performed using equation (1) to obtain values between 0 and 1 for each field.

$$Z_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \tag{1}$$

where *Z* represents the normalized value (Normalized IoU:0.3 in Table 1) and *X* represents the IoU value of 0.3.

Table 1: Accuracy measurements based on IoU 0.3 and performance ranking

CNN	IoU:0.3	Normalized IoU:0.3	Rank
Faster R-CNN with Resnet 101	95.44	1	1
Mobilenet v.1	85.01	0	5
Retinanet with Resnet 50	91.94	0.621	3
SSD with Mobilenet v.2	85.9	0.085	4
YOLOv3 with Darknet-53	91.8	0.65	2

3 Data

The training set was collected using a crowdsourced image-labelling service called LabelBox and consists of 13 images of 5,472 x 3,648 pixels each; the total label count was 18,469. The survey mission took place in November 2018 in Bosque del Apache Wildlife Refuge, New Mexico (see Figure 2) using a DJI Mavic sUAS equipped with a Hasselblad L1D-20c RGB sensor operated at a flight altitude of 40 m. There were 13 species classes in the dataset: American wigeon, mallard, northern pintail, other, Canada goose, teal, sandhill crane, gadwall northern shoveller, ring-necked duck, redhead, ruddy and snow goose. However, the gadwall northern shoveller, ring-necked duck, redhead, ruddy and snow goose had fewer than 200 labels each. They were therefore not representative and, compared to other larger classes, had little chance of being detected. They would have been a source of uncertainty to the CNN. The remaining 8 species were aggregated to 3 waterfowl types: duck, goose and crane.

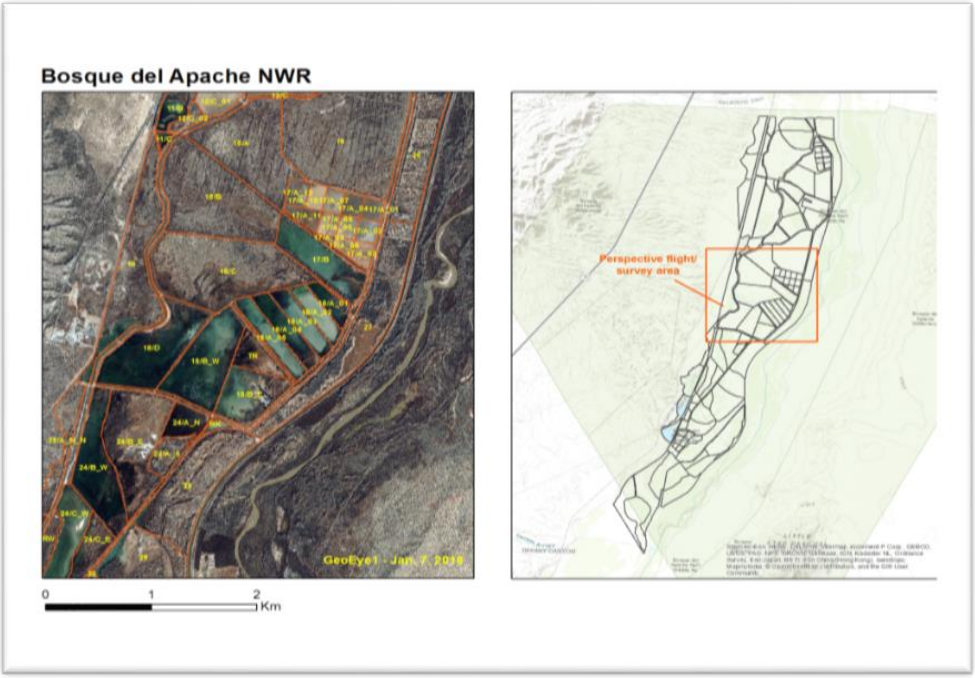

Figure 2: Location of Bosque del Apache Wildlife Refuge

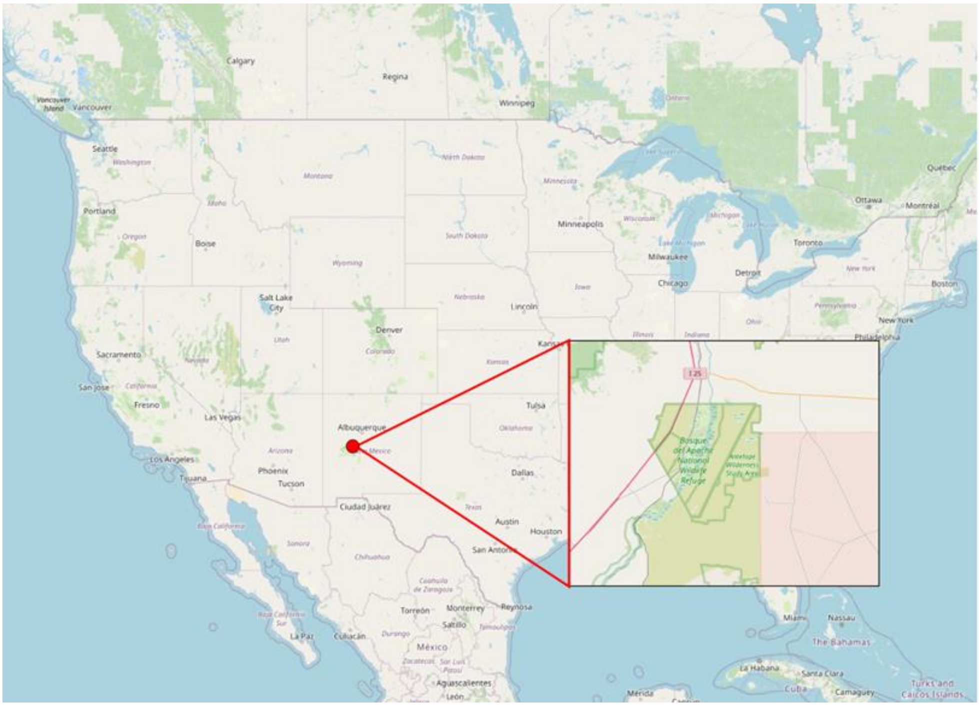

Figure 3: Location of Bosque del Apache Wildlife Refuge

4 Pre-processing

Pre-processing comprises the following steps:

- Convert LabelBox JSON Annotations to format accepted by each CNN.
- Split each image to a group of sub-images for the object to be easily identified.
- Remove multiple labels so that each target object has only one label.
- Construct a validation set (images not fed to the CNN in the training process).

The data are then segmented into three parts, namely training, testing and evaluation. The conversion process resulted in 18,469 labels for the 8 classes. However, as the dataset was labelled by 13 experts, redundant labels for each object had to be removed. If there were two or more labels with an IoU of more than 50%, the label with the larger area was removed (Figure 4).

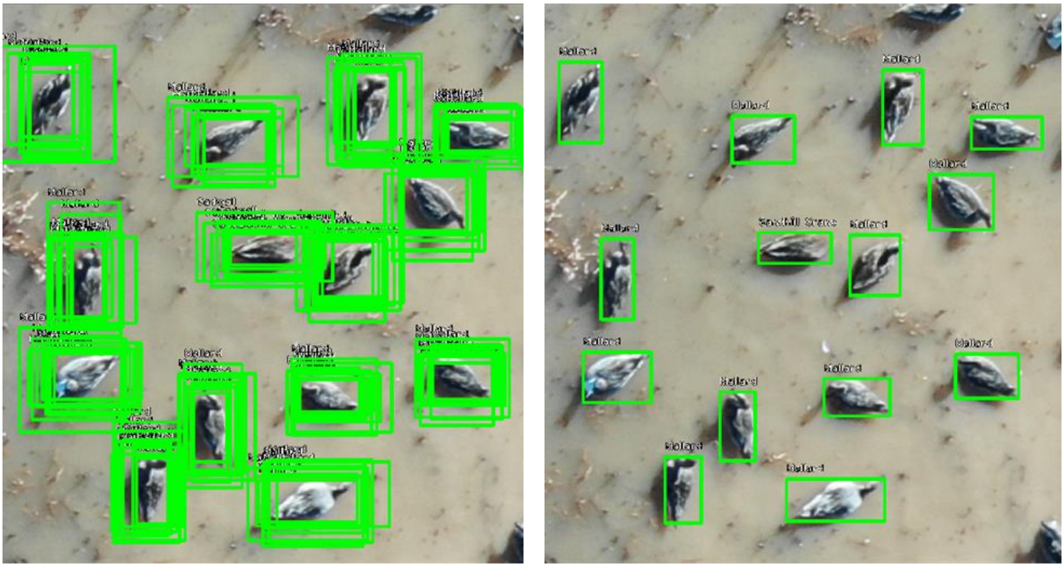

Figure 4: Removing multiple labels from the training set

Each image has 5,472 x 3,648 pixels and the average label size is 52 x 54 pixels, occupying 0.014% of the total image area. This small percentage limits the ability of YOLOv3 to detect desired objects, causing the CNNs to perform very poorly on the original images (<3% accuracy of count). It was necessary to crop to multiple sub-images in order to enlarge the ratio of the area covered by the label in the image. Each image was therefore tiled to 56 sub-images (7 rows, 8 columns) of 684 x 521 pixels, increasing the average percentage of the area covered by a single label to 0.78%. Figure 4 shows how a bird is enlarged by the cropping procedure.

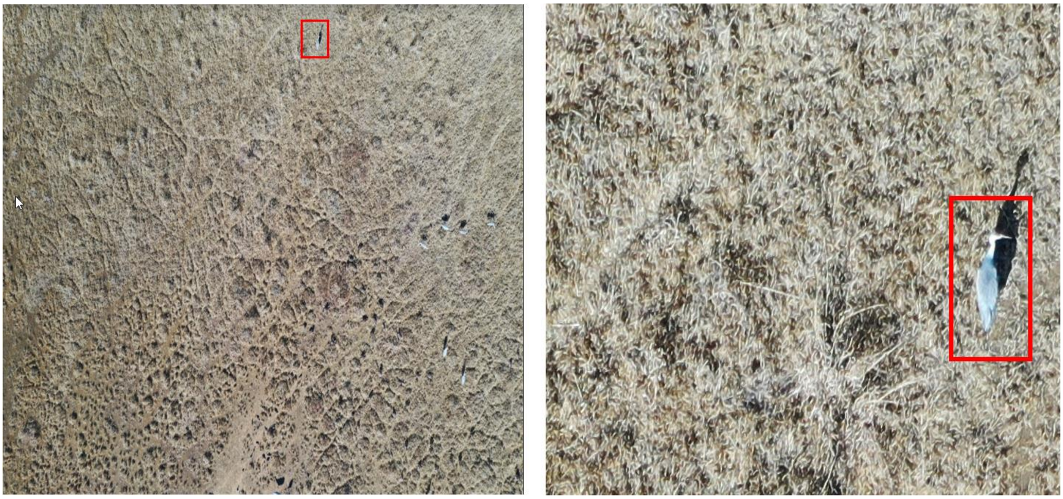

Figure 5: Image of the same bird shown before and after cropping

5 Model training and evaluation process

Table 2 presents the training parameters of the CNNs.

Table 2: Training Parameters for the CNNs

Training Parameter	YOLOv3	Retinanet	Faster R-CNN
Number of training samples	2908	2908	2908
Epochs (species/fowl type/detection)	16/12/11	3/2/2	118/92/87
Training time (sec./epoch)	212	1340	2280

Accuracy measurement is divided into accuracy of *detection* and accuracy of *identification*. The accuracy of detection is defined by how close the number of successful predictions is to the actual number of birds in the test data. The identification accuracy refers to how close the number of successful class predictions is to the actual number of class occurrences in the test dataset. Nevertheless, manual evaluation of the results (looking at each individual image result) is necessary to understand why an error has occurred in the detections. The most common method of representing prediction results is to build a confusion matrix. A cross-validation process was implemented by taking the ground truth labels (10% of the dataset that contains all waterfowl classes and different background conditions) as a reference, where the difference between the IoU of each label in the ground truth and prediction labels was calculated. The prediction label with the highest IoU was then assigned to the ground truth label. A successful count is defined as a prediction label that intersects a labelled object with the same class in the ground truth. This assessment was carried out for all levels (detection, fowl type and species) to explore the effect of increasing the number of classes on the general counting performance.

6 Results

The results of CNN for waterfowl detection, waterfowl-type identification and species identification were evaluated by performing a cross-validation, where each prediction label is compared to the corresponding ground truth label. Finally, the results were summarized in a confusion matrix.

6.1 Detection

Figure 6 shows the detection performance in relation to the 3 hierarchical levels for the 3 CNN implementations. The x-axis represents implementation levels for each CNN (1: detection; 3: waterfowl type; 8: species); the y-axis represents the detection performance of the predictions broken down into three categories: detected waterfowl; undetected waterfowl; detection of non-waterfowl objects as waterfowl. Increasing the number of classes to be detected limits the ability of CNNs to detect waterfowl, and increases the number of undetected waterfowl (omission errors) and the number of non-waterfowl species detected (commission errors).

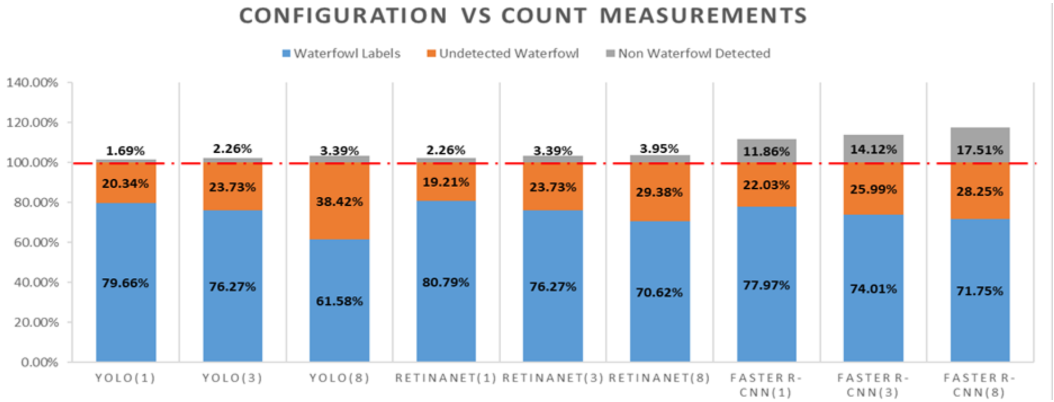

Figure 6: Detection Performance (Detection levels)

6.2 Waterfowl Type

The second assessment concerned the detection and identification accuracy of the CNNs on 3 classes of waterfowl (Duck, Goose and Crane); the confusion matrices in Tables 3–5 document the detection accuracies. The grey cells indicate successful classification (true-positive prediction); the rows represent the reference or the ground truth; and the columns present the measurement modelled.

Table 3: YOLO confusion matrix for fowl type (true-positives in grey)

Class	Duck	Goose	Crane	undetected	Ground Truth Total	Non-WF
Duck	95	14	0	34	143	3
Goose	1	21	0	7	29	1
Crane	0	0	4	1	5	0
Detection Total	96	35	4	42	177	4

Table 4: Retinanet confusion matrix for fowl type (true-positives in grey)

Class	Duck	Goose	Crane	undetected	Grand Truth Total	Non-WF
Duck	106	4	0	33	143	6
Goose	4	17	1	7	29	0
Crane	0	0	3	2	5	0
Detection Total	110	21	4	42	177	6

Table 5: Faster R-CNN confusion matrix for fowl type (true-positives in grey)

Class	Duck	Goose	Crane	undetected	Grand Truth Total	Non-WF
Duck	106	3	0	34	143	25
Goose	8	10	0	11	29	0
Crane	1	0	3	1	5	0
Detection Total	115	13	3	46	177	25

Duck was confused with Goose but not Crane by all CNNs, probably as a result of the very different size and coloration of cranes compared to ducks and geese.

6.3 Species level

The third assessment was for accuracy of species detection and identification. The confusion matrices (Tables 7–9) reveal how the CNNs behaved at species level. The grey cells represent successful classification (true-positive predictions). As the dataset contains six out of the eight classes belonging to fowl-type 'duck', it is no surprise that the classes belonging to this fowl-type were very frequently confused with each other (unlike what happened for the classes Canada Goose and Sandhill Crane).

Table 6: YOLO confusion matrix for species (true-positives in grey)

Class	American wigeon	Canada goose	Gadwall	Mallard	Northern pintail	Other	Sandhill crane	Teal	Undetected	GT total	Non-WF
American wigeon	0	1	0	0	2	1	0	0	4	8	0
Canada goose	0	28	0	0	0	0	1	0	0	29	4
Gadwall	0	2	0	0	1	0	0	1	2	6	0
Mallard	0	8	0	13	23	4	0	1	46	95	0
Northern pintail	0	2	0	1	2	0	1	1	2	9	1
Other	0	3	0	0	5	0	0	0	10	18	0
Sandhill crane	0	1	0	0	0	0	4	0	0	5	1
Teal	0	0	0	0	3	0	0	0	4	7	0
Detection Total	0	45	0	14	36	5	6	3	68	177	6

Table 7: Retinanet confusion matrix for species (true-positive in grey)

Class	American wigeon	Canada goose	Gadwall	Mallard	Northern pintail	Other	Sandhill crane	Teal	Undetected	GT total	Non-WF
American wigeon	2	0	2	0	0	0	0	2	2	8	3
Canada goose	0	20	1	0	0	1	1	1	5	29	1
Gadwall	2	0	0	0	0	1	0	1	2	6	1
Mallard	24	0	14	10	5	1	0	13	28	95	0
Northern pintail	3	0	0	0	1	2	0	0	3	9	0
Other	4	0	1	2	0	1	0	3	7	18	0
Sandhill crane	0	0	0	0	0	0	3	0	2	5	0
Teal	1	0	3	0	0	0	0	0	3	7	2
Detection Total	36	20	21	12	6	6	4	20	52	177	7

Table 8: Faster R-CNN confusion matrix for species (true-positives in grey)

Class	American wigeon	Canada goose	Gadwall	Mallard	Northern pintail	Other	Sandhill crane	Teal	Undetected	GT total	Non-WF
American wigeon	0	0	1	3	3	0	0	1	0	8	13
Canada goose	2	12	1	0	0	4	0	4	6	29	0
Gadwall	0	0	1	1	0	2	0	1	1	6	2
Mallard	4	1	2	13	25	10	0	12	28	95	0
Northern pintail	0	0	0	1	0	4	0	1	3	9	1
Other	1	0	0	1	0	5	0	1	10	18	15
Sandhill crane	1	0	0	0	0	0	3	0	1	5	0
Teal	0	0	0	0	3	2	0	1	1	7	0
Detection Total	8	13	5	19	31	27	3	21	50	177	31

7 Discussion

The first observation on the results is that the number of waterfowl detected decreased as we increased the number of classes for detection (waterfowl level to sub-species level). This is probably due to the decrement in the number of training examples per waterfowl type as we segregated waterfowl species. At species and fowl-type levels, the CNN tries to capture class-specific patterns using fewer training samples, defining each class independently. The second observation is that the result images often include individual waterfowl that are labelled multiple times (see Figure 3). This is probably due to the limited ability of the CNN to distinguish between different waterfowl species. This limitation can be most directly addressed by including more samples for each class. The third insight is the effect of surroundings and population density on performance. It was clear that better performance is achieved in low population densities and clear surroundings (no shadows, shrubs etc.), suggesting that there will be a sample bias based on the landscape composition being observed.

The confusion matrices shown above have a non-traditional setup; the number of detected objects does not match the number of objects in the ground truth. Some objects were not detected, and others were detected that did not represent waterfowl. This forced us to add undetected and non-waterfowl classes which have no ground truth reference. Our results were therefore reported in terms of correct/incorrect classification, undetected waterfowl, and non-waterfowl detection.

8 Conclusion and Future work

The results of this research show that CNNs have the potential to automate the process of waterfowl detection and identification. Across all CNN architectures, using a higher number of classes to train the network resulted in lower waterfowl detection accuracy, increasing both omission and commission errors. Increasing the number of classes to be detected reduced the number of labels generated by YOLO and Retinanet, but not Faster R-CNN.

Our experience suggests that this potential depends on the volume, quality, and structural and textual composition of training data in terms of image resolution and labelling correctness. We discovered that hyperspatial image data of waterfowl captured by a sUAS have special characteristics when compared to many other CNN applications: target objects are small while species morphology is often relatively similar. These characteristics need to be taken into consideration for adequate pre-processing.

In this project, valuable insights were collected on the performance and applicability of several freely accessible CNNs to waterfowl detection and identification using hyperspatial airborne imagery. One of the most important steps in future research will be to investigate how a much larger waterfowl training dataset (hundreds of thousands of training samples) will affect the results and behaviour of the CNNs. The findings presented here suggest that a phased approach should be explored, whereby a single-class 'waterfowl' detector is followed by a waterfowl-labelling step using a CNN trained on all classes. Finally, the class imbalance between species and fowl types is likely to be a persistent issue even as the number of labels available for training increases. Given the performance with respect to rare classes of the models evaluated here, and the potential importance of rare classes in a wildlife survey context, data augmentation techniques with the potential to magnify the training samples of those rare classes warrant investigation.

From a practical perspective, the possibility of entering aerial images of waterfowl to the CNN in their original size also warrants further research. Keeping images in their original size would reduce the pre-processing tasks, making the model more efficient, but the computational feasibility of doing this needs to be investigated. For example, what would the memory requirements be? Could the computations be achieved through modifying state-of-the-art CNNs? Or would a CNN architecture have to be built from scratch, allowing the design of the hyperparameters to be oriented towards this specific application?

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).

Abd-Elrahman, A., Pearlstine, L., & Percival, F. (2005). Development of pattern recognition algorithm for automatic bird detection from unmanned aerial vehicle imagery. Surveying and Land Information Science, 65(1), 37.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). Ieee.

Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 0588-0592). IEEE.

Chabot, D., & Francis, C. M. (2016). Computer-automated bird detection and counts in high-resolution aerial images: A review. Journal of Field Ornithology, 87(4), 343-359.

Chen, K., Loy, C. C., Gong, S., & Xiang, T. (2012, September). Feature mining for localised crowd counting. In BMVC (Vol. 1, No. 2, p. 3).

Ghorbanzadeh, O., Blaschke, T., Gholamnia, K., Meena, S. R., Tiede, D., & Aryal, J. (2019). Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. Remote Sensing, 11(2), 196.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Grenzdörffer, G. J. (2013). sUAS-based automatic bird count of a common gull colony. International archives of the photogrammetry, Remote sensing and spatial information sciences, 1, W2.

Hong, S. J., Han, Y., Kim, S. Y., Lee, A. Y., & Kim, G. (2019). Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. Sensors, 19(7), 1651.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Krogh, A. (2008). What are artificial neural networks?. Nature biotechnology, 26(2), 195-197.

Laliberte, A. S., & Ripple, W. J. (2003). Automated wildlife counts from remotely sensed imagery. Wildlife Society Bulletin, 362-371.

Linchant, J., Lisein, J., Semeki, J., Lejeune, P., & Vermeulen, C. (2015). Are unmanned aircraft systems (UAS s) the future of wildlife monitoring? A review of accomplishments and challenges. Mammal Review, 45(4), 239-252.

van Gemert, J. C., Verschoor, C. R., Mettes, P., Epema, K., Koh, L. P., & Wich, S. (2014). Nature conservation drones for automatic localization and counting of animals. In European Conference on Computer Vision (pp. 255-270). Springer, Cham.

Wang, H., & Raj, B. (2017). On the origin of deep learning. arXiv preprint arXiv:1702.07800.

Wilson, R. P., Culik, B., Danfeld, R., & Adelung, D. (1991). People in Antarctica—how much do Adélie Penguins Pygoscelis adeliae care?. Polar biology, 11(6), 363-370.

Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. (2015). Exploiting image-trained CNN architectures for unconstrained video classification. arXiv preprint arXiv:1503.04144.