Science and Technology Publishing (SCI & TECH)
ISSN: 2632-1017
Vol. 10 Issue 1, January - 2026

Analysis of an Onboard MobileNetV3-Small Model for UAV-Based Bird Classification Using Aerial Imagery

Chisom S. Nwokonko

Department of Electrical and Electronic Engineering,
Imo State University, Owerri

Abstract

This paper presents a lightweight onboard bird classification model for unmanned aerial vehicle (UAV) imagery using MobileNetV3-Small. The work addresses the need for species-level image classification on resource-constrained UAV sensor boards, where continuous cloud processing may be slow, costly, or unavailable. Bird images from the CUB-200-2011 and NABirds datasets were used to form a balanced six-class classification task with 333 samples per class in the reported evaluation. Image patches were preprocessed and passed through a MobileNetV3-Small classifier with global average pooling, dropout, a dense classification head, and softmax output. The training loss reduced from about 2.3 to 0.3, while the validation loss reduced from about 2.5 to 0.4 over 40 epochs. The model achieved average precision of 95.96%, average recall of 96.28%, and average F1-score of 96.12%. The average notable misclassification count was 4.80. These results show that MobileNetV3-Small provides strong classification performance for UAV-based bird recognition. The available experimental record did not include actual species names, the exact train-validation split ratio, or board-level inference measurements; these items are therefore treated as study limitations and are recommended for future reporting.

Keywords: Unmanned aerial vehicle, MobileNetV3-Small, onboard classification, bird classification, resource-constrained device, CUB-200-2011, NABirds.

1. Introduction

Unmanned aerial vehicles are increasingly used for environmental monitoring, wildlife observation, precision agriculture, and smart farm surveillance. Their ability to capture images from low altitude and flexible viewpoints makes them useful for detecting bird activity across farms, wetlands, coastlines, and other open environments. In agricultural settings, early identification of pest birds can support timely deterrence and reduce crop damage.

Reliable onboard bird classification is important when a UAV must respond immediately without depending on continuous cloud connectivity. Practical UAV sensor boards have limited processing power, memory, battery energy, and communication bandwidth. A classification model for this setting should therefore be accurate, compact, and computationally efficient. Heavy image classification networks may give strong recognition accuracy, but they are often unsuitable for real-time inference on small airborne platforms.

MobileNetV3-Small is a suitable candidate for this task because it was designed for low-resource mobile and embedded inference. It combines depthwise separable convolution, inverted residual bottlenecks, squeeze-and-excitation attention, and hardware-aware design choices. These features reduce computational cost while preserving useful visual representations for image classification.

The research gap addressed in this work is the need for a clear lightweight classification stage for UAV-based bird monitoring. Many bird classifiers focus mainly on recognition performance, while practical UAV use also requires attention to onboard execution, decision support, and the limitations of embedded hardware. This study focuses on MobileNetV3-Small as a single lightweight onboard classifier. Comparison with heavier or alternative lightweight classifiers is reserved for future work.

The contribution of this paper is threefold. First, it presents a coherent UAV onboard bird classification workflow based on MobileNetV3-Small. Second, it reports the preserved classification results from the six-class bird image experiment. Third, it adds system and model architecture diagrams to clarify the implementation pathway from image capture to onboard decision support.

2. Related Work

Fine-grained bird recognition requires models that can identify subtle inter-class differences, including feather texture, beak structure, wing contour, color pattern, and body shape. Datasets such as CUB-200-2011 and NABirds have supported many studies on fine-grained visual categorization because they provide bird species labels and image annotations. These datasets are useful for benchmarking classification models before deployment in field conditions.

UAV-based monitoring has also become important in wildlife observation and agricultural surveillance. UAV platforms can cover large areas quickly, but their imagery often contains small objects, complex backgrounds, motion blur, and changing illumination. These conditions make the classification stage more difficult than ordinary close-range image classification. As a result, an onboard model should be accurate enough for decision support and compact enough for embedded execution.

Convolutional neural networks have achieved strong performance in image classification tasks, but standard models can be computationally expensive for onboard UAV use. Lightweight networks address this limitation by reducing parameter count and multiply-accumulate operations. MobileNetV3 improves earlier MobileNet designs through hardware-aware neural architecture search, NetAdapt optimization, squeeze-and-excitation, and efficient nonlinearities. These properties make the MobileNetV3-Small variant appropriate for UAV sensor boards where inference speed and energy efficiency are important.

Recent UAV vision studies also show that lightweight CNN designs remain important when image understanding must run on low-cost or embedded aerial platforms. For example, lightweight CNN models have been applied to UAV-captured video classification and UAV-based image classification, with emphasis on reduced complexity, fast inference, and practical edge deployment [15], [16].

Existing UAV vision systems often use a two-stage perception process. The first stage locates or crops the target object, while the second stage classifies the cropped image region. This study follows that practical design and focuses on the onboard classification stage. The emphasis is on classification performance and the expected suitability of the classifier for resource-constrained UAV hardware. This paper is not a detection paper; it focuses on the classification stage after a bird region has been cropped or selected.

3. Materials and Methods

3.1 Dataset Description

The study used bird images from the CUB-200-2011 and NABirds datasets. These datasets are widely used for fine-grained bird image classification. The available result summary shows that the final evaluation was organized into six balanced representative classes, denoted as Class_0 to Class_5. Each class had 333 samples in the evaluation summary, giving balanced support for the per-class performance assessment.

The result tables do not state whether the CUB-200-2011 and NABirds images were fully merged or whether selected species were sampled from both datasets. The available results therefore support a six-class balanced classification experiment, but the exact species selection rule, duplicate-removal procedure, dataset merging strategy, and train-validation split ratio should be supplied from the original data preparation log. The balanced class support reduces bias during metric interpretation because precision, recall, and F1-score can be compared directly across the six classes. In a practical UAV setting, the classes may represent bird species or species groups that are relevant to monitoring or pest management. The original class identifiers are retained because the result tables used anonymized class labels. Future dataset documentation should replace Class_0 to Class_5 with the actual bird species names when these names are available.

Dataset reporting limitation: Replication of this study requires the original dataset preparation log. That log should identify the selected species, the source dataset for each image, any duplicate-removal rule used when combining datasets, the train-validation split ratio, and the random seed used for splitting. These details do not change the reported results, but they are needed for full reproducibility.

3.2 System Model

The onboard classification system begins with aerial image capture from the UAV camera. Each frame is screened and preprocessed through resizing and normalization. A bird region is then cropped from the detected object area and passed to the MobileNetV3-Small classifier. The classifier outputs the predicted bird class and confidence score. The UAV decision module can use this result to ignore a harmless class, deter a pest bird, pursue additional observation, or transmit the event log to mission control.

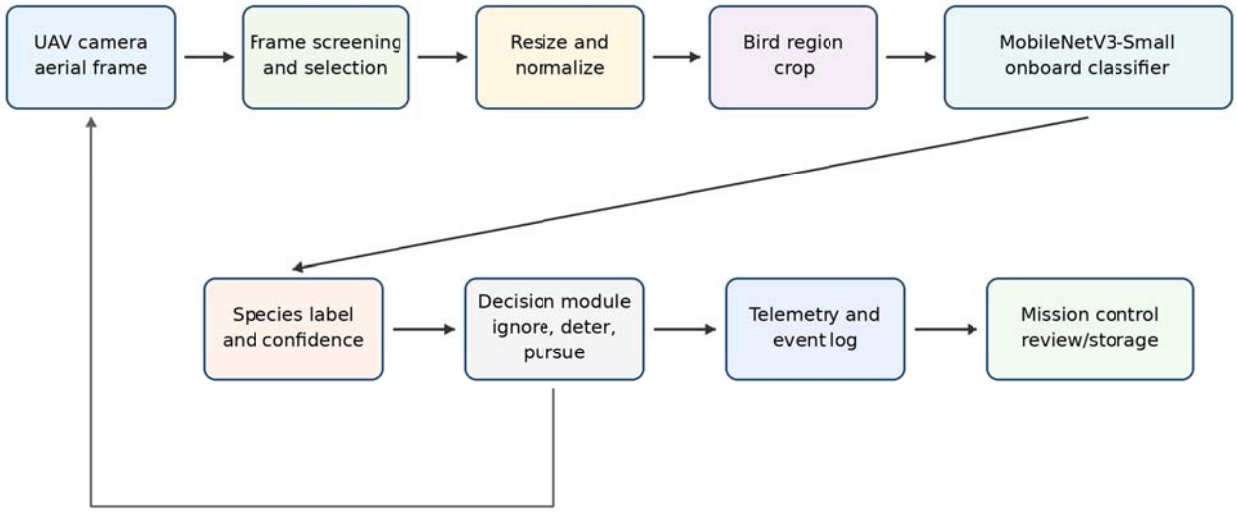

Figure 1: UAV onboard bird classification system architecture showing image capture, preprocessing, bird-region cropping, MobileNetV3-Small classification, confidence estimation, and decision support.

The architecture in Figure 1 shows the operational pathway from UAV image acquisition to onboard decision support. The feedback link allows the UAV to capture additional image frames when the confidence score is low or when the target remains within the field of view.

3.3 Image Preprocessing and Data Splitting

Each input image patch was resized to the input dimension required by the MobileNetV3-Small classifier. Pixel values were normalized to improve numerical stability during training. Data augmentation was applied to improve robustness to UAV imaging conditions. The augmentation operations included small rotations, horizontal flips, brightness adjustment, mild zooming, and translation. These operations simulate field variations caused by UAV motion, view angle, illumination, and target scale.

The dataset was divided into training and validation subsets. The training subset was used to update the network weights, while the validation subset was used to monitor generalization during the training process. The exact train-validation ratio was not available in the reported training record and should be added from the original training log before final submission. The loss curve covers 40 epochs. The close movement of the training and validation losses across the epochs indicates that the model did not overfit severely.

3.4 Experimental Setup

The experimental setup was organized around the MobileNetV3-Small classifier and the balanced six-class bird image subset. Table 1 separates the settings available in the reported record from the items that require confirmation from the original training log.

Table 1: Experimental setup for the onboard MobileNetV3-Small classifier.

Item	Setting / reporting status
Dataset source	CUB-200-2011 and NABirds bird image datasets
Dataset construction	Six-class balanced subset; exact merging and species-selection rule requires original training log
Number of evaluated classes	Six balanced anonymized classes, labelled Class_0 to Class_5

Support per class	333 samples per class in the reported result table
Input form	Cropped bird image patch
Input size	224 x 224 x 3 image patch for MobileNetV3-Small
Preprocessing	Resize, normalization, and data augmentation
Train-validation split	Requires original training log
Training duration	40 epochs, based on the loss curve
Batch size	Requires original training log
Optimizer	Requires original training log
Learning rate	Requires original training log
Weight decay / scheduler	Requires original training log
Training framework	Requires original training log
Classifier head	Global average pooling, dropout, dense layer, and softmax
Loss function	Categorical cross-entropy
Evaluation metrics	**Precision, recall, F1-score, loss trend, and misclassification count**

The table improves reproducibility because it shows which parts of the experiment are supported by the available record and which parts should be supplied from the training log. The manuscript therefore avoids unsupported claims about optimizer choice, learning rate, batch size, training framework, or measured embedded latency.

3.5 MobileNetV3-Small Classification Model

The cropped bird image patch is passed to MobileNetV3-Small for feature extraction and species-level classification. The model uses compact convolutional layers and bottleneck blocks to extract discriminative visual features from the bird patch. The final feature map is converted to a vector by global average pooling, regularized using dropout, and passed to a dense classification layer. A softmax function then produces the probability of each bird class.

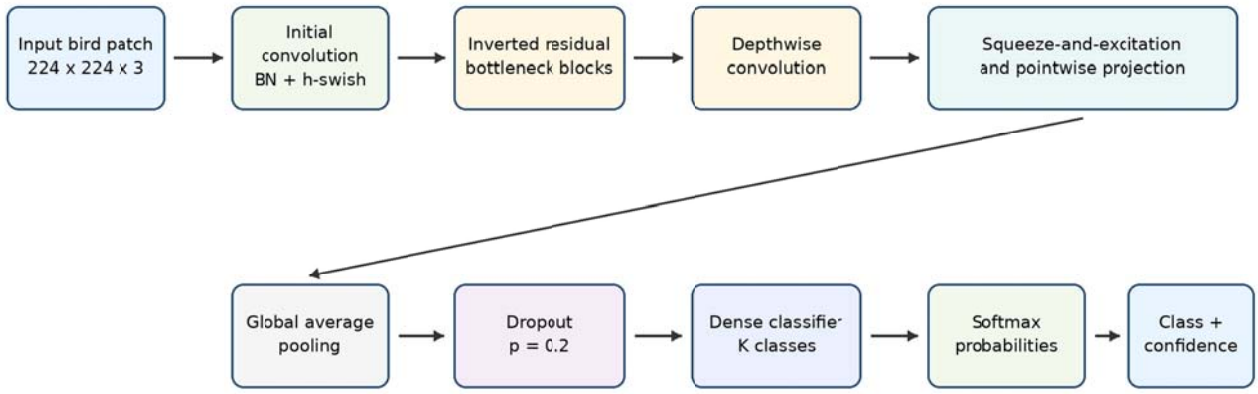

Figure 2: MobileNetV3-Small architecture for onboard bird classification showing the input image patch, compact feature extraction blocks, global average pooling, dropout, dense classification layer, and six-class softmax output.

Figure 2 shows the MobileNetV3-Small pathway used in the study. The input patch is processed through an initial convolution, inverted residual bottleneck blocks, depthwise convolution, squeeze-and-excitation attention where applicable, and pointwise projection. The global average pooling and softmax classifier then convert the extracted features into class probabilities.

For an input bird patch R, the feature embedding produced by the backbone is expressed in (1).

$$f = \text{MobileNetV3-Small}(R) \quad (1)$$

The convolutional transformation in a generic layer is represented in (2).

$$F_l = \sigma\big(\text{BN}(W_l * F_{l-1} + b_l)\big) \quad (2)$$

where F_l is the output feature map of layer l, W_l is the convolution kernel, b_l is the bias term, BN(.) is batch normalization, σ(.) is the activation function, and □ denotes convolution. After feature extraction, global average pooling produces a compact vector:

$$z = \mathrm{GAP}(f) \quad (3)$$

The logits for K bird classes are obtained using (4).

$$o = W_c z + b_c \quad (4)$$

The class probability distribution is obtained using the softmax expression in (5).

$$p(y = k \mid R) = \frac{\exp(o_k)}{\sum_{j=1}^{K} \exp(o_j)} \quad (5)$$

The predicted class is selected using (6).

$$\hat{y} = \underset{k}{\mathrm{argmax}} p(y = k \mid R) \quad (6)$$

The training objective is the categorical cross-entropy loss in (7).

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(p_{ik}) \quad (7)$$

Knowledge distillation can be included as an optional extension if a teacher model and distillation setting are specified. It is not interpreted as part of the reported experiment because no teacher model or distillation ablation result was provided. The optional objective is:

$$L = \alpha L_{CE} + (1 - \alpha) T^2 \, \mathrm{KL}(q_t \parallel p_t) \quad (8)$$

where α is the loss balancing factor, T is the distillation temperature, KL(.) is the Kullback-Leibler divergence, q_t is the teacher distribution, and p_t is the student distribution. The results in this paper are interpreted only for the reported MobileNetV3-Small classifier because no teacher model was specified in the available experimental record.

3.6 Training and Inference Procedure

Algorithm 1 summarizes the onboard bird classification training and inference procedure.

Algorithm 1: Onboard MobileNetV3-Small Bird Classification Pipeline

Input: Cropped bird image patch R and class set K

Output: Predicted bird species label $\hat{y}$ and confidence score c

1. Resize R to the MobileNetV3-Small input size.
2. Normalize pixel values and apply augmentation during training.
3. Pass R through the MobileNetV3-Small backbone to obtain feature map f.
4. Apply global average pooling to produce the vector z.
5. Apply dropout with $p = 0.2$ during training.
6. Compute logits o using the dense classification layer.
7. Convert logits to class probabilities using softmax.
8. Compute categorical cross-entropy loss and update model weights during training.
9. During inference, select $\hat{y} = \mathrm{argmax}_k p(y = k \mid R)$.
10. Return $\hat{y}$ and $c = \max_k p(y = k \mid R)$ to the onboard decision module.

4. Results and Discussion

4.1 Training and Validation Loss

The reported loss result shows a stable learning process. Training loss reduced from about 2.3 to 0.3, while validation loss reduced from about 2.5 to 0.4 over 40 epochs. The two curves follow a similar downward trend.

This indicates that the model learned useful discriminative patterns from the bird images without a clear sign of severe overfitting.

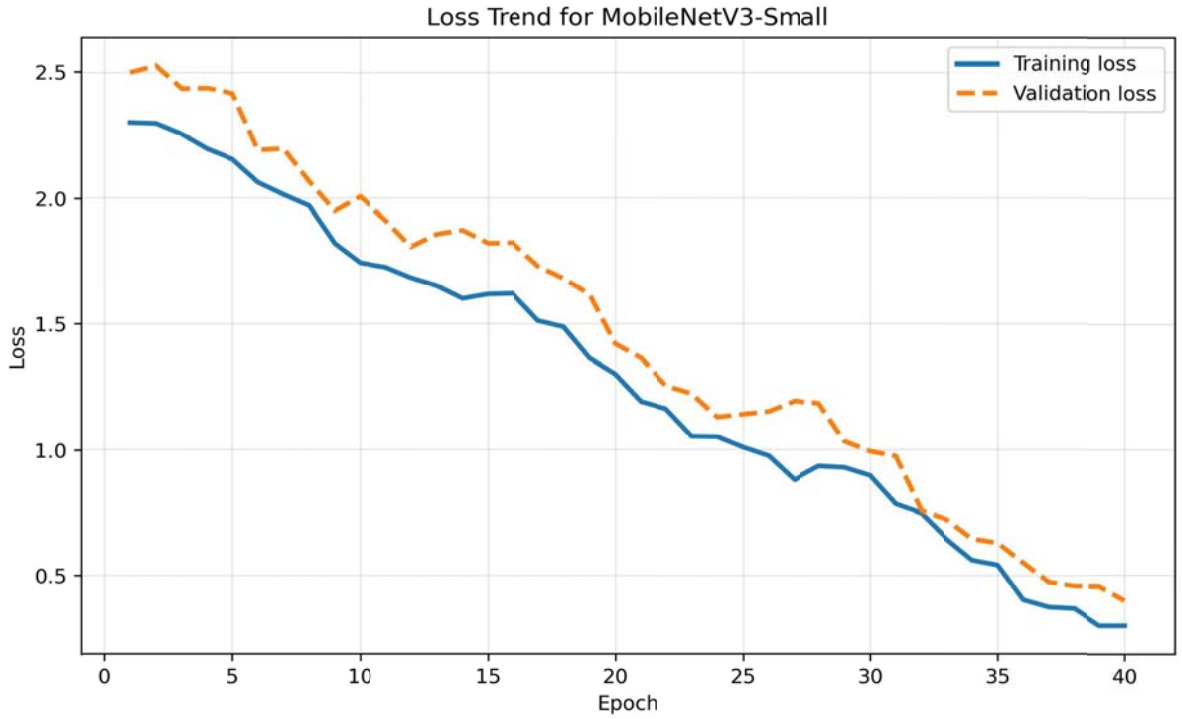

Figure 3: Loss versus epoch trend for the MobileNetV3-Small onboard bird classification model.

4.2 Per-Class Classification Performance

The reported per-class performance is presented in Table 2. The class names are kept as Class_0 to Class_5 because the available experimental record did not provide the actual bird species names. If the original dataset preparation log is later available, these anonymized labels should be replaced with the corresponding species names. The average precision, recall, and F1-score are computed across the six balanced classes.

Table 2: Per-class result for the onboard MobileNetV3-Small classification model.

Class	Support	Precision (%)	Recall (%)	F1-Score (%)
Class_0	333	95.80	96.40	96.10
Class_1	333	96.20	95.90	96.05
Class_2	333	96.85	97.10	96.97
Class_3	333	96.10	96.40	96.25
Class_4	333	94.30	95.10	94.70
Class_5	333	96.50	96.80	96.65
AVERAGE	**333.00**	**95.96**	**96.28**	**96.12**

The precision value of 95.96% means that most of the images predicted as a given bird class were correct. The recall value of 96.28% means that the model retrieved most true samples belonging to each class. The F1-score of 96.12% confirms that the balance between precision and recall is strong. Accuracy is not reported because the available results did not provide a separate overall accuracy value.

4.3 Misclassification Analysis

The notable misclassification result from the study is shown in Table 3. The highest confusion occurred between Class_1 and Class_4, with 9 cases where Class_1 was predicted as Class_4 and 6 cases where Class_4 was predicted as Class_1. Smaller confusion values occurred between Class_2 and Class_3, Class_3 and Class_2, and Class_5 and Class_0.

Table 3: Notable misclassification summary for the onboard MobileNetV3-Small model.

True Class	Predicted Class	Misclassification Count

Class_1	Class_4	9
Class_4	Class_1	6
Class_2	Class_3	4
Class_3	Class_2	3
Class_5	Class_0	2
AVERAGE		**4.80**

The misclassification pattern suggests that visually similar bird classes remain the main source of error. Such errors are expected in fine-grained classification because related species or visually similar species groups may share feather color, body contour, and wing structure. Since the results use anonymized class labels, the analysis cannot name the exact confused species. In later field work, replacing the anonymous labels with actual bird species names will make the error analysis more useful for agricultural and ecological decision-making.

4.4 Deployment Suitability and Limitations

The reported performance supports the use of MobileNetV3-Small as a lightweight classifier for UAV-based bird classification. The architecture is compact and was designed for low-resource inference, while the reported precision, recall, and F1-score are high. However, the result set did not include measured inference time, memory use, battery impact, model file size, RAM use, frame rate, or the name of the embedded UAV board. The deployment interpretation is therefore limited to architectural suitability and expected embedded efficiency, not measured board-level performance.

Table 4: Deployment-relevant profile of the onboard classifier.

Deployment item	Interpretation
Model type	MobileNetV3-Small lightweight CNN classifier
Target environment	Resource-constrained UAV onboard sensor board
Reported classification evidence	Average precision = 95.96%, recall = 96.28%, F1-score = 96.12%
Model size	Requires original training log
Number of parameters	Requires original training log
Inference time per image	Requires board-level test
RAM use / FPS / energy use	Requires board-level test
Practical implication	**Suitable for further onboard testing, but hardware validation is still required**

In pest bird management, the UAV can use the predicted class and confidence score to decide whether to continue monitoring, activate a deterrence response, or log the event for later review. Any deterrence response should be non-lethal, species-aware, and compliant with local wildlife protection rules. The compact architecture is expected to support lower latency, reduced energy use, and improved autonomy when compared with heavier classification models, but these expectations remain subject to hardware validation and should be backed by board-level measurements.

5. Conclusion

This paper presented a UAV onboard bird classification study using MobileNetV3-Small. The system model architecture and MobileNetV3-Small model architecture were included to clarify the pathway from UAV image capture to onboard decision support. The model reduced training loss from about 2.3 to 0.3 and validation loss from about 2.5 to 0.4 over 40 epochs. It achieved average precision of 95.96%, average recall of 96.28%, and average F1-score of 96.12%, with an average notable misclassification count of 4.80. These results show strong classification performance for the six-class bird image task. The main limitation is that the available experimental record does not include actual species names, exact split ratio, complete training hyperparameters, or real-time inference measurements on a physical UAV embedded board. Future work should validate the model on actual UAV hardware, report model size and inference speed, expand the number of bird species, and replace anonymized class labels with real species names.

References

[1] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

[2] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Technical Report CNS-TR-2011-001, 2011.

[3] G. Van Horn et al., "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 595-604, doi: 10.1109/CVPR.2015.7298658.

[4] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 7132-7141, doi: 10.1109/CVPR.2018.00745.

[7] M. C. Hayes et al., "Drones and deep learning produce accurate and efficient monitoring of large-scale seabird colonies," Ornithological Applications, vol. 123, no. 3, 2021, doi: 10.1093/ornithapp/duab022.

[8] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A Survey," IEEE Geoscience and Remote Sensing Magazine, vol. 10, no. 1, pp. 91-124, 2022, doi: 10.1109/MGRS.2021.3115137.

[9] G. Tang, Y. Li, X. Ding, and Y. Zhang, "A Survey of Object Detection for UAVs Based on Deep Learning," Remote Sensing, vol. 16, no. 1, 2024, Art. no. 149, doi: 10.3390/rs16010149.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[11] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in Proc. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.324.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 4700-4708, doi: 10.1109/CVPR.2017.243.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[15] N. A. Othman and I. Aydin, "Development of a Novel Lightweight CNN Model for Classification of Human Actions in UAV-Captured Videos," Drones, vol. 7, no. 3, Art. no. 148, 2023, doi: 10.3390/drones7030148.

[16] X. Deng, M. Shi, B. Khan, Y. H. Choo, F. Ghaffar, and C. P. Lim, "A lightweight CNN model for UAV-based image classification," Soft Computing, vol. 29, pp. 2363-2378, 2025, doi: 10.1007/s00500-025-10512-3.