Science and Technology Publishing (SCI & TECH)
ISSN: 2632-1017
Vol. 10 Issue 3, March - 2026

UAV-Based Mobilenetv3 Lightweight Detector For Real-Time Bird Detection In Rice Farms

Chisom S. Nwokonko
Department of Electrical and Electronic Engineering,
Imo State University, Owerri

Abstract
This paper presents a UAV-based MobileNetV3 lightweight detector for real-time bird detection in rice farms. The work addresses bird detection from high-resolution drone imagery using a resource-constrained onboard processing platform. The detector combines a MobileNetV3 feature extraction backbone with lightweight one-stage detection configurations suitable for SSD-style and YOLO-Lite-style deployment. The Distant Bird Detection Dataset was used. It contains 47,260 images and 60,971 manually annotated bird instances at 3840 x 2160 pixel resolution, covering hawk, crow, and wild bird categories. The evaluated onboard detector achieved mAP@0.5 of 55.05%, mAP@0.5:0.95 of 23.27%, precision of 69.96%, recall of 61.40%, and F1-score of 65.41%. It also achieved 35.1 ms inference latency, 28.5 FPS, 0.013 J energy per inference, and 58.1 MB memory footprint. These results show that the model is suitable as a lightweight first-stage onboard alert detector, although further work is required to reduce false alarms and improve strict localization accuracy.

Keywords: Unmanned aerial vehicle, MobileNetV3, lightweight detector, one-stage detection, bird detection, rice farms, edge AI, low-latency inference.

1. Introduction

Bird damage is a major field-management problem in rice production because large groups of birds can feed on rice grains during sensitive growth and maturity stages. Traditional bird-scaring approaches depend heavily on manual patrol, stationary scare devices, noise makers, nets, or periodic human intervention. These approaches are often expensive, inconsistent, labor-intensive, and difficult to scale across large farms. UAV-based monitoring offers a practical alternative because drones can move over wide field areas, observe bird activity from above, and support rapid deterrent response when birds are detected.

The use of UAVs for bird detection presents a difficult computer vision task. Birds usually occupy a very small portion of a drone image, especially when the camera captures high-resolution scenes from a safe flight altitude. The rice-field background may also contain vegetation texture, lighting variation, shadows, field boundaries, and moving objects that can confuse the detector. The detection model must therefore localize small targets while still running in real time on embedded onboard processors. A large detector may produce better accuracy, but it may also consume too much memory, power, and inference time for sustained drone operation.

Lightweight convolutional neural networks provide a useful solution to this problem. MobileNetV3 is designed for mobile and embedded devices and uses depthwise separable convolution, squeeze-and-excitation attention, and efficient nonlinear activations. When paired with one-stage object detectors such as SSD and YOLO-Lite, it can process images without the region proposal stage used in two-stage detectors. This design reduces inference delay and supports deployment on drone-borne computing units such as Raspberry Pi boards, neural accelerators, or Coral TPU modules.

This study presents a UAV-based MobileNetV3 lightweight detection framework for real-time bird detection in rice farms. The work focuses on practical onboard deployment using a compact feature extractor and one-stage

detection configurations. The goal is to provide a fast first-stage alert model that can support rice-field monitoring under limited memory, power, and processing resources.

The main contributions of the paper are as follows: (i) an edge-oriented UAV bird detection framework for rice-field monitoring is presented; (ii) MobileNetV3 is adopted as a compact feature extraction backbone for onboard inference; (iii) SSD and YOLO-Lite detection heads are described as alternative one-stage localization modules; (iv) the system is evaluated using the reported DBD dataset results; and (v) the accuracy, latency, memory, power, and energy tradeoffs are interpreted in relation to real-time UAV deployment.

2. Related Work

Object detection has moved from hand-crafted features to deep learning models capable of learning discriminative spatial features from image data. Two-stage detectors generally provide strong localization performance but often require greater computational resources. One-stage detectors, including SSD and YOLO-family variants, predict object classes and bounding boxes in a single forward pass, making them attractive for embedded vision. SSD predicts bounding-box offsets and class scores from multiple feature maps, while YOLO-style models divide the image into grids and predict boxes, objectness, and class probabilities directly.

Mobile networks have become important in edge AI because they reduce the computational cost of convolutional networks. MobileNetV3 combines neural architecture search, depthwise separable convolution, and squeeze-and-excitation channel attention to improve the accuracy-latency balance. This makes it suitable for UAV applications where onboard battery, memory, thermal, and processing constraints are strict. The present work builds on this lightweight design and adapts it to bird detection in rice farms.

Small-object detection remains a major limitation in drone imagery. Objects captured from high altitude may be visually small, blurred, partially occluded, or difficult to distinguish from background texture. These limitations explain why a detector can obtain acceptable IoU for matched boxes but still record low AP values under strict multi-threshold evaluation. The present study therefore interprets the results as an edge-efficiency baseline rather than as a final high-accuracy pest-bird detector.

Recent surveys on lightweight object detection for edge devices confirm that compact detectors must balance accuracy, latency, memory footprint, and power consumption. Recent aerial-image small-object detection studies also emphasize that small target size, viewpoint changes, complex background, scale variation, and orientation variation make UAV object detection more difficult than ordinary ground-level image detection [13], [14]. These findings support the design choice in this study, where MobileNetV3 is used as a compact onboard baseline while the results are interpreted cautiously because of the observed scale-wise AP limitation.

3. Materials and Methods

3.1 Dataset Description

The Distant Bird Detection Dataset used in this study contains 47,260 images and 60,971 manually annotated bird instances. The images have a resolution of 3840 x 2160 pixels. The annotated bird categories are hawk, crow, and wild bird. Each annotation provides bounding-box information that supports object localization during training and evaluation.

The resizing step improves speed but also compresses small bird targets. This tradeoff is important for interpreting the results. The model can run at 28.5 FPS with low memory and energy demand, but the strict mAP@0.5:0.95 result remains low because small target localization becomes more difficult after spatial down-sampling.

The exact training, validation, and testing split was not available in the retained experimental record. Therefore, this paper reports the available aggregate evaluation results without introducing an assumed data split.

3.2 Proposed System Model

The proposed UAV-based bird detection system consists of six main stages: image capture, pre-processing, lightweight feature extraction, one-stage detection, post-processing, and decision support. The reported results correspond to a single onboard MobileNetV3 lightweight detector evaluated as one compact SSD/YOLO-Lite detection configuration. The SSD head represents the multi-scale anchor-based localization path, while the YOLO-Lite head represents the grid-based low-latency deployment path. The paper does not claim a separate comparison between the two heads because separate SSD and YOLO-Lite training logs were not available. This clarification is important because the retained accuracy and energy values should be interpreted as aggregate onboard-detector results.

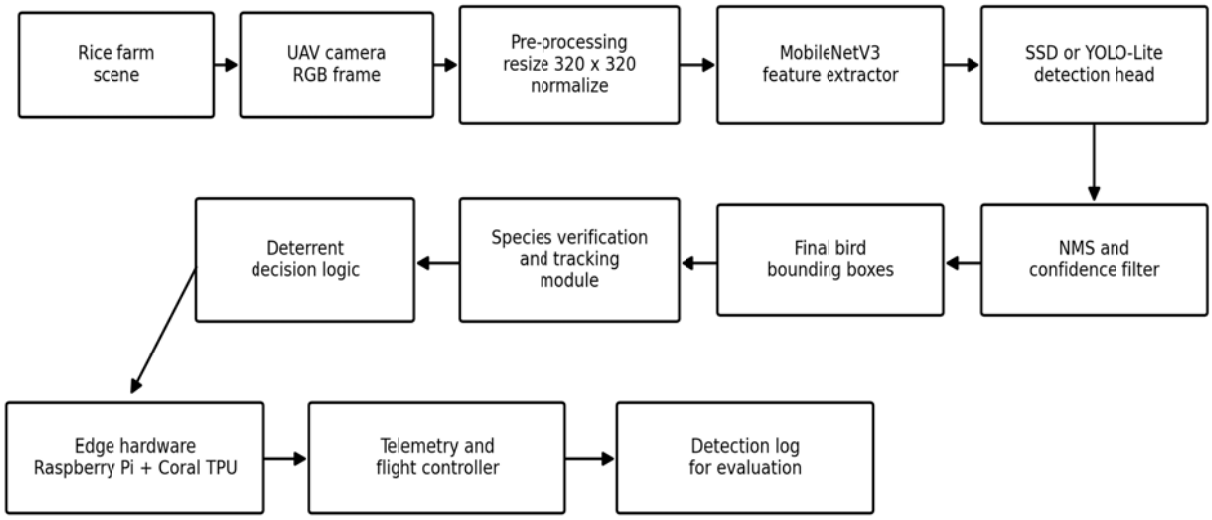

Figure 1. System model architecture of the UAV-based MobileNetV3 bird detection framework.

3.3 MobileNetV3 Feature Extraction Backbone

The MobileNetV3 backbone is the main feature extraction component of the proposed detection framework. It replaces standard convolution with depthwise separable convolution to reduce the number of multiply-accumulate operations. A standard convolution with kernel size $K \times K$, input channels M, output channels N, and feature-map size $H \times W$ requires approximately $K^2 \times M \times N \times H \times W$ operations. Depthwise separable convolution reduces this cost to $K^2 \times M \times H \times W + M \times N \times H \times W$ by separating spatial filtering from channel mixing.

For an input tensor X, the depthwise convolution for channel m can be expressed as:

$$Y_m(i,j) = \sum_u \sum_v W_m(u,v) X_m(i+u, j+v) \quad (1)$$

The pointwise 1 x 1 convolution then combines channel-wise outputs as:

$$Z_n(i,j) = \sum_m P_{n,m} Y_m(i,j) \quad (2)$$

The squeeze-and-excitation mechanism improves channel selectivity by applying global average pooling followed by nonlinear channel gating. For channel c, the global descriptor is:

$$s_c = \frac{1}{HW} \sum_i \sum_j Z_c(i,j) \quad (3)$$

The channel gate is computed as:

$$a = \sigma\big(W_2 \,\mathrm{ReLU}(W_1 s)\big) \quad (4)$$

The final reweighted channel output is given by:

$$\hat{Z}_c = a_c Z_c \quad (5)$$

This operation helps the detector emphasize bird-like visual patterns such as wing edges, body contours, and contrast changes while suppressing irrelevant background texture.

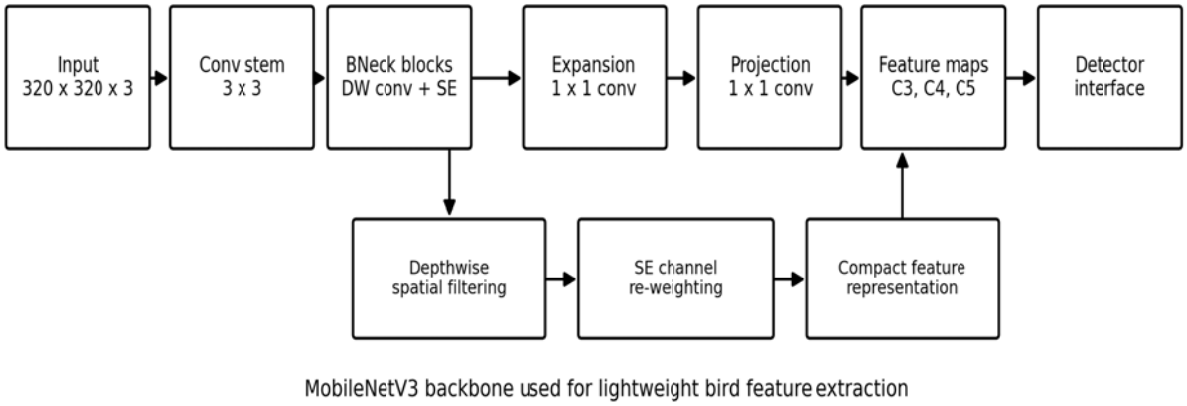

Figure 2. MobileNetV3 backbone architecture used for lightweight feature extraction.

3.4 MobileNetV3-SSD Detection Model

The SSD variant uses MobileNetV3 feature maps at different scales to detect birds of different apparent sizes. At each feature-map location, SSD predicts offsets for a set of default anchor boxes and class confidence scores. The detector output for a matched default box can be represented as:

$$D = \{(b_i, c_i, p_i)\}_{i=1}^{N} \qquad (6)$$

where b_i is the predicted bounding box, c_i is the predicted class label, p_i is the confidence score, and N is the number of retained detections after post-processing.

The SSD training objective combines classification loss and localization loss as:

$$L_{\text{SSD}} = \frac{1}{N_m}[L_{\text{conf}}(c, c^*) + \alpha L_{\text{loc}}(b, b^*)] \qquad (7)$$

where N_m is the number of matched default boxes, L_conf is the confidence loss, L_loc is the localization loss, alpha is the balancing parameter, c and b are predicted class and box outputs, and c* and b* are the corresponding ground-truth values.

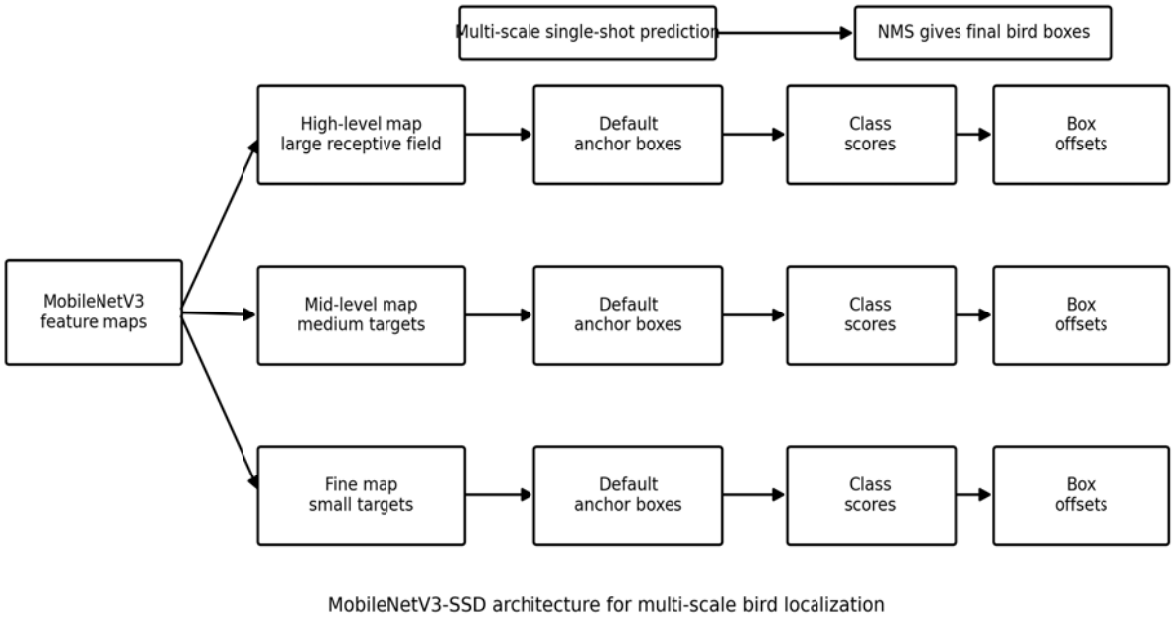

Figure 3. MobileNetV3-SSD model architecture for multi-scale bird localization.

3.5 MobileNetV3-YOLO-Lite Detection Model

The YOLO-Lite path is included as a low-latency one-stage detection configuration for the same MobileNetV3 backbone. It performs object localization and classification using grid-based predictions and fewer computational operations. This design is useful when the UAV platform has stricter memory, power, or processing limits.

$$Y \in \mathbb{R}^{S \times S \times B(5+C)} \quad (8)$$

where S is the grid size, B is the number of predicted boxes per cell, and C is the number of classes. Each predicted box contains x, y, w, h, objectness, and class-probability values. The compound YOLO-Lite loss can be summarized as:

$$L_{\text{YOLO}} = \lambda_{\text{coord}} L_{\text{box}} + L_{\text{obj}} + \lambda_{\text{noobj}} L_{\text{noobj}} + L_{\text{cls}} \quad (9)$$

where L_{box} penalizes localization error, L_{obj} penalizes incorrect objectness prediction, L_noobj controls false detections in empty cells, and Lcls penalizes class-prediction error. Channel pruning and quantization-aware training are included to make the YOLO-Lite path deployable on low-power drone hardware.

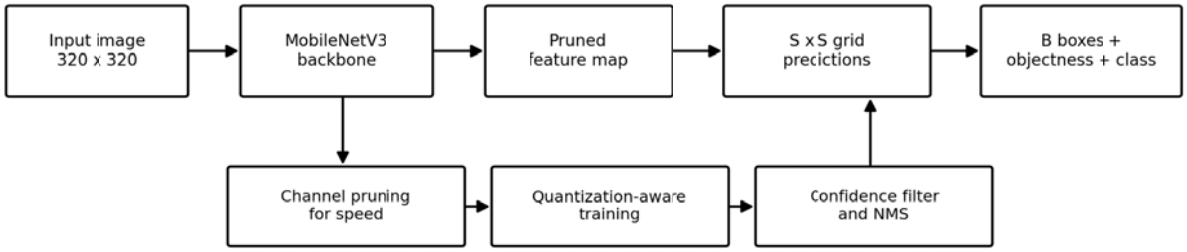

Figure 4. MobileNetV3-YOLO-Lite model architecture for low-latency detection.

3.6 Training and Evaluation Procedure

The training procedure begins with image resizing, normalization, annotation conversion, and mini-batch preparation. All images are resized to 320 x 320 pixels to reduce computation and support onboard inference. The exact training, validation, and testing split was not available in the retained experimental record. Therefore, this paper reports the available aggregate evaluation results without introducing an assumed data split. The SSD and YOLO-Lite paths are treated as lightweight one-stage detector configurations within the same MobileNetV3 edge-deployment design, while the retained experimental record reports one aggregate onboard detector.

$$L_{\text{total}} = L_{\text{backbone}} + \beta L_{\text{det}} \quad (10)$$

where L_det is either L_SSD or L_YOLO depending on the selected detection head, and beta controls the contribution of the detection loss. After each training epoch, validation loss is computed to observe convergence and possible overfitting.

Model evaluation uses mAP@0.5, mAP@0.5:0.95, AP_small, AP_medium, AP_large, mean IoU, precision, recall, F1-score, false positive rate, false negative rate, inference latency, FPS, memory footprint, and energy per inference. The SSD and YOLO-Lite paths are treated as lightweight one-stage detector configurations within the same MobileNetV3 edge-deployment design, while the retained experimental record reports one aggregate onboard detector.

4. Results and Discussion

4.1 Training Behavior of the Onboard MobileNetV3 Detection Model

The training and validation loss curves show that the model learned from the dataset and reached a stable operating point. The training curve shows that loss dropped from about 3.5 to about 0.3, while the validation loss dropped from about 3.7 to about 0.5. The two curves remain reasonably close across the training process, which suggests that the evaluated detector did not suffer from severe overfitting. However, the loss behavior alone does not prove strong detection performance, so it must be interpreted together with the mAP, precision, recall, F1-score, false-positive, and deployment results.

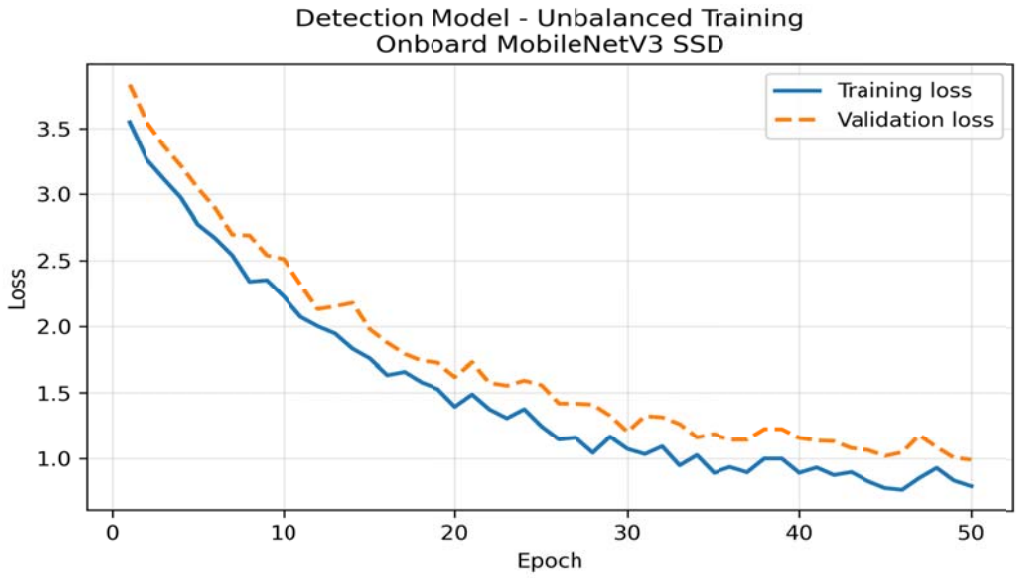

Figure 5. Loss versus epoch plot for the onboard MobileNetV3 bird detection model.

4.2 Detection Accuracy Results

The detection accuracy results are presented in Table 1. They correspond to the evaluated onboard MobileNetV3 lightweight detector. The values should be interpreted as aggregate results for the edge-detection configuration, not as separate benchmark results for SSD and YOLO-Lite.

Table 1. Detection accuracy results for the evaluated onboard MobileNetV3 lightweight bird detector.

Metric	**Value**
mAP@0.5	55.05%
mAP@0.5:0.95	23.27%
AP_small	0.00%
AP_medium	0.00%
AP_large	0.00%
Mean IoU (matched boxes)	0.740
Precision	69.96%
Recall	61.40%
F1-score	65.41%
False Positive Rate (FPR)	0.835, equivalent to 83.5% if interpreted as a proportion
False Negative Rate (FNR)	38.60%

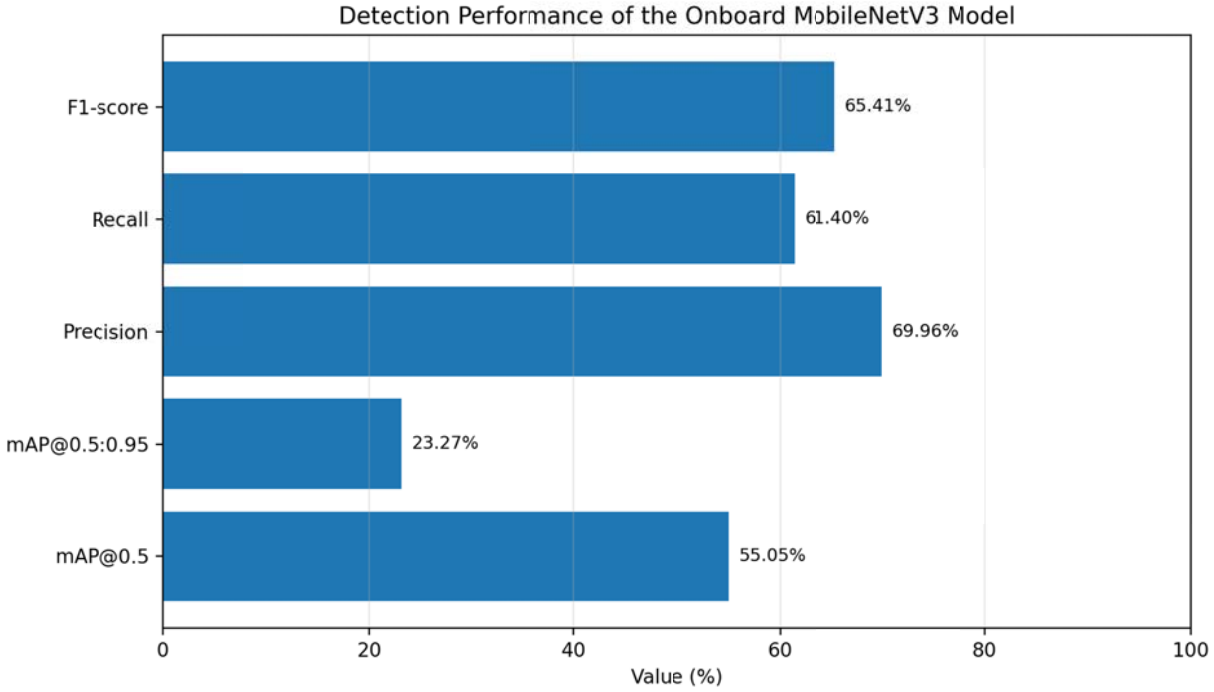

Figure 6. Summary of detection metrics from the onboard MobileNetV3 model.

The precision value of 69.96% shows that many positive detections corresponded to bird objects. The recall value of 61.40% indicates that the detector still missed a substantial number of birds. The F1-score of 65.41% confirms moderate overall detection performance. The mAP@0.5 of 55.05% is acceptable for a lightweight first-stage alert model, but the mAP@0.5:0.95 of 23.27% shows weak localization accuracy under stricter IoU thresholds. The zero AP values for small, medium, and large objects are major limitations. Further evaluation is needed to determine whether they resulted mainly from detector weakness, object-size mapping, anchor mismatch, or evaluation-configuration limitations.

The false positive rate of 0.835 is also important. If interpreted as a proportion, it is equivalent to 83.5% and indicates a high false-alarm tendency. This means that the detector may wrongly identify background objects as birds under some UAV imaging conditions. Therefore, the model is better used as a first-stage onboard alert detector than as a final autonomous decision model.

The dataset contains hawk, crow, and wild bird classes, but class-wise precision, recall, F1-score, and AP values were not available in the retained result record. Therefore, the paper reports aggregate detection results only. Future work should include class-wise analysis to identify which bird categories are more difficult for the onboard detector.

4.3 Edge Deployment and Energy Efficiency Results

The power and energy results are shown in Table 2. They describe the same evaluated MobileNetV3 lightweight detector and should not be interpreted as separate SSD and YOLO-Lite deployment results.

Metric	Evaluated onboard detector value
Inference Power (W)	1.35
Energy per Inference (J)	0.013
Average FPS	28.5
Inference Latency (ms)	35.1
CPU Utilization (%)	9.8
Memory Footprint (MB)	58.1
Model Size (MB)	12
Estimated Battery Drain (% / min)	1.1
Flight Time Impact (relative)	Lowest

Table 2. Power consumption and energy-efficiency results for the evaluated onboard MobileNetV3 lightweight bird detector.

The edge-deployment results are the main strength of the model. The inference latency of 35.1 ms supports near real-time frame processing while the drone is moving across the farm. The average speed of 28.5 FPS is suitable for lightweight monitoring tasks, and the 0.013 J energy-per-inference value indicates low energy demand. The 58.1 MB memory footprint and 12 MB model size also support deployment on embedded platforms such as Raspberry Pi-class computers with accelerator support.

4.4 Harmonized Interpretation of the Results

The experimental results show that the detector has a clear accuracy-efficiency tradeoff. It does not yet provide strong detection accuracy under strict mAP evaluation, but it is computationally light enough for onboard UAV operation. The high false positive rate and zero scale-wise AP values limit its use as a standalone decision system. Its strongest value is fast screening of drone video frames before further verification.

A practical farm-deployment system should therefore use this detector as an early warning component. The lightweight model can rapidly screen video frames on the UAV, while a stronger verification stage can reduce false alarms and improve confidence before deterrent action is taken.

5. Conclusion

This paper presented a UAV-based MobileNetV3 lightweight detection framework for real-time bird detection in rice farms. The model was designed for onboard inference under limited memory, processing, and energy resources. The evaluated detector achieved mAP@0.5 of 55.05%, mAP@0.5:0.95 of 23.27%, precision of 69.96%, recall of 61.40%, and F1-score of 65.41%. It also achieved 35.1 ms inference latency, 28.5 FPS, 0.013 J energy per inference, and 58.1 MB memory footprint. These results show that the detector is useful as a lightweight first-stage alert model. However, the zero scale-wise AP values and high false-positive rate show that it still requires improvement before being used as a fully reliable autonomous bird-detection system.

References

[1] S. Fujii, K. Akita, and K. Sato, "Distant bird detection for safe drone flight and its dataset," in Proc. 17th Int. Conf. Machine Vision Applications (MVA), 2021, pp. 1-5. [Online]. Available: https://www.mva-org.jp/Proceedings/2021/papers/O1-1-3.pdf

[2] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[4] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132-7141, doi: 10.1109/CVPR.2018.00745.

[5] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Computer Vision - ECCV 2016, Lecture Notes in Computer Science, vol. 9905. Cham, Switzerland: Springer, 2016, pp. 21-37, doi: 10.1007/978-3-319-46448-0_2.

[6] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers," in Proc. IEEE Int. Conf. Big Data, 2018, pp. 2503-2510, doi: 10.1109/BigData.2018.8621865.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[8] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in Computer Vision - ECCV 2014, Lecture Notes in Computer Science, vol. 8693. Cham, Switzerland: Springer, 2014, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.324.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," Int. J. Comput. Vis., vol. 88, no. 2, pp. 303-338, 2010, doi: 10.1007/s11263-009-0275-4.

[11] Google Coral, "Edge TPU documentation," Google, 2024. [Online]. Available: https://coral.ai/docs/edgetpu/

[12] TensorFlow, "TensorFlow Lite guide," Google, 2024. [Online]. Available: https://www.tensorflow.org/lite

[13] P. Mittal, "A comprehensive survey of deep learning-based lightweight object detection models for edge devices," Artificial Intelligence Review, vol. 57, article 242, 2024, doi: 10.1007/s10462-024-10877-1.

[14] W. Hua and Q. Chen, "A survey of small object detection based on deep learning in aerial images," Artificial Intelligence Review, vol. 58, article 162, 2025, doi: 10.1007/s10462-025-11150-9.