Proceedings of the 2022 Sustainable Research and Innovation Conference
JKUAT Main Campus, Kenya
5 - 6 October, 2022

Improving Small Pest Bird Detection in YOLOv5s for Autonomous Bird Deterrent Systems

Jeff Mboya[1], Steve Nyaga[2], Jackson Njiri[3], and Shohei Aoki[4]

Abstract— Granivorous birds are known to destroy grain crops in farms, and various studies are underway to find a solution to the problem. In recent studies, state-of-the-art deep learning technologies have been actively applied. However, image resolution has made detecting smaller pest birds a challenging task. Moreover, high-speed and low flight altitude bring in the motion blur on the densely packed birds, which leads to great challenge of object distinction. For that purpose, this paper presents an improved YOLOv5s model based on the YOLOv5 single-stage detector. The improved YOLOv5s model is proposed for application in bird deterrent systems where image background noise is high and identification of small birds is poor. To achieve this, the CSPDarknet backbone in YOLOV5s was replaced with DenseNet. Three convolution blocks and modules of the CSPbottleneck in YOLOV5s were also replaced with Transformer encoder blocks, and PANet in the original YOLOV5s neck was substituted with BiFPN. To further improve the performance of the improved YOLOv5s model, one additional prediction head was introduced for tiny object detection in the head. Both the original YOLOv5s and improved YOLOv5s models were trained using images from the Klim dataset. The dataset contains 1607 images for training, 340 images for validation, and another 357 images for testing. The test results on the Klim dataset showed an improvement of up to 4.8% in mean average precision when detecting smaller birds with the improved YOLOv5s at 50% Intersection Over Union, at the cost of just a 4 milliseconds increase in inference time. Based on a comparison with the original YOLOv5s model on the Klim dataset, the proposed YOLOv5s model outperformed the original model and achieved the highest performance in terms of accuracy (97.30%), area under receiver operating characteristic curve (93.78%), precision (98.54%), and F1-score (57.85%). The results showed that the modified YOLOv5s model is suitable for detecting small birds in various environments and consequently applicable in bird deterrent systems.

Keywords— Deep Learning, Object Detection, Small Pest Birds, YOLOv5s[1]

I. INTRODUCTION

Small object detection in images can be difficult owing to a low resolution of an object detection model and environmental variables [1]. Most existing systems that employ object detection do so at real-time speeds, requiring particular computing capacity, notably if the computation is to take place on the same hardware that acquires the pictures [2]. This is true for many bird deterrent systems [3, 4].

Because of the basis of object detection, the features of smaller birds lose relevance as each layer of the object detection model processes them. In this work, "small objects" refers to objects that have fewer pixel points in the picture. YOLOv5 is a single-stage object detector that is renowned for its effectiveness and responsiveness [5]. YOLOv5 is available in four models, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, each of which offers different detection accuracy and performance. Respectively the YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x mean small (s), medium (m), large (l), and extra-large (x) models. Each model has its pros and cons, but generally, the differences are in their complexities, performances, and overall accuracy. Experimental results show the size of the models varies from 14MB to 168MB, mAP values for the full COCO dataset range from 36.8 to 50.1, and the inference time on the Nvidia V100 GPU varies from 2.2 to 6.0 milliseconds. These results are suitable for real-time usage and are, however, achieved with a dedicated architecture.

Generally, YOLOv5 has a simple and versatile architecture that is easily deconstructed, modified, and rebuilt. Although YOLOv5 is a powerful tool, it is not optimized to detect small objects because it is intended to be a general-purpose object detector [6, 7]. However, a lot of the systems that use the model and try to optimize it mostly depend on changing certain variables or expanding the training dataset to increase performance, with little regard for architectural modifications to adapt it for a particular use case [8-10].

This study makes recommendations for improving the performance of YOLOv5s model in terms of small object recognition, with significant real-world applications. In this paper, an automated bird deterrent system is specifically taken into consideration. We will explore the effects of various modifications and offer a modified YOLOv5s model capable of detecting small birds better while retaining real-time processing

[1]Jeff Mboya, Department of Mechatronic Engineering, JKUAT
(email: mboya.jeff@students.jkuat.ac.ke, phone: +254732723906)
[2]Steve Nyaga, Department of Mechatronic Engineering, JKUAT
[3]Jackson Njiri, Department of Mechatronic Engineering, JKUAT
[4]Shohei Aoki, Department of Mechatronic Engineering, JKUAT

speeds. The remainder of this article is organized as follows: Section II outlines related works. Section III gives a brief review of the original YOLOv5s model, and the improved YOLOv5s model is described in section IV. Section V describes data preprocessing and training. Section VI describes the experimental results and section VII concludes the paper.

II. RELATED WORK

Real-time object detection systems have gained popularity due to the need for them to fulfill current needs. For instance, autonomous bird deterrent systems use a number of cameras and image processing techniques to detect approaching birds and scare them away. In manufacturing, identifying faulty assembly parts is necessary. The two examples given above demonstrate how important real-time object detection is. However, in order to be used later as inputs for other activities, such as triggering visual or acoustic bird deterrents as in autonomous bird deterrent systems, such real time object detection systems need early object detection. Early detection makes it such that representations of objects are often small [11].

The main objective of small object detection is typically to quickly identify objects in an image, especially those objects that are small in size [12]. This implies that objects of interest are either those that have a physically large appearance but only take up a small portion of the image, like trains, or are actually small in appearance, like plates and computer mouse, as shown in Fig. 1. Therefore, detecting small objects is a difficult problem in computer vision because, in addition to having small depictions of the objects, the work is additionally complicated by the variety of input pictures [13]. For instance, a bird picture may be in a variety of resolutions, and if the resolution is low, the object detector may not be able to detect the birds. The visual information needed in this situation to localize the birds will be severely constrained. Small objects can also be deformable or be covered by larger ones [14].

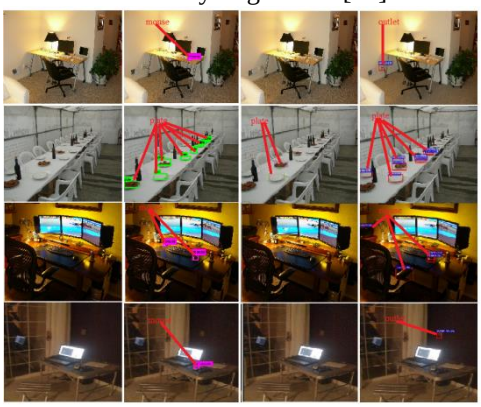

Fig. 1 Illustration of small objects [14]

To address the problems of small object detection, a variety of models have been put out with their accompanying assessments. However, the suggested detectors focus their abilities on the identification of objects of all sizes, not only small ones. For instance, Wang et al. [15] propose a YOLOv4 network that applies a spatial pyramid pooling layer to extract features and compute the features over an entire image regardless of image size. The authors show that the YOLOv4 object detection neural network, based on the CSP approach, scales both up and down and is applicable to small and large networks while maintaining optimal speed and accuracy. However, the model has a bias towards bigger objects.

R-CNN [16] improves on the prior techniques in various ways. For one, a picture is shrunk to a predefined size and fed into the network, which then uses an external algorithm to detect small objects. Fast R-CNN [17], an improvement on [16], uses areas of interest to extract a fixed-length feature from the proposal of each feature map. Instead of utilizing an external network, faster R-CNN [17] employs its own network to detect small objects. R-CNN and Fast R-CNN, among other convolution neural network-based object detectors can be divided into: 1) single-stage detectors: YOLOX [12], FCOS [13], Scaled-YOLOv4 [15], and EfficientDet [16]. 2) two-stage detectors: CenterNet2 [16] and VFNet [17]. 3) anchor-based detectors: ScaledYOLOv4 [15] and YOLOv5 [18]. 4) anchor-free detectors: CenterNet [16], YOLOX [12], and RepPoints [19]. But from the perspective of architecture, they generally consist of two parts, a CNN-based backbone for image feature extraction, and a detection head to predict the class and bounding box for the objects. Furthermore, object detectors built in recent years frequently insert several layers between the backbone and the head, known as the neck.

So far, most detection models do well on the MS COCO and PASCAL VOC datasets [18]. The datasets contain objects that take up medium or large portions of an image that contains a few small objects, resulting in an imbalance of data across objects of different sizes, leading to a bias of models towards bigger objects [19]. Furthermore, MS COCO and PASCAL VOC datasets have fewer small object classes, and most of the state-of-the-art detectors, both in one-stage and two-stage approaches, have struggled with detecting the small objects. Consequently, there have been attempts to enhance the detection of small objects [20], but several of these initiatives concentrate on focusing image processing on a particular region of the picture [21-23] or on two-stage object detectors, which are recognized for improving performance at the expense of inference time and are thus less suitable for real-time systems [24,25]. This is also the rationale for the proliferation of single-stage object detectors for such application [26]. Another apparent option explored to get around the problem is to raise the resolution of input image, however doing so significantly lengthens process cycle time [27].

As a consequence, the aim of this article is to improve the performance of YOLOv5s model in terms of small object recognition, with the objective of application in automated bird deterrent system. The following are the contributions of this paper: (1) A YOLOv5s modified model created specifically for

improved small object detection. (2) Developing a methodology for modifying the architecture of YOLOv5s in order to increase performance in a specific task.

I. *YOLOv5 Network Module*

YOLOv5 is a single-stage object detection algorithm proposed by Glenn Jocher in 2020 [28]. Based on variances in network width and depth, YOLOv5 is separated into four network model variations: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [29]. The YOLOv5s network has the quickest computation speed but the lowest average precision, whereas the YOLOv5x network has the opposite properties. Fig. 2 depicts the entire design of the YOLOv5 model, which includes the backbone, neck, and head. Typically, YOLOv5 uses the architecture of CSPDarknet53, with SPP layer as the backbone, PANet as neck, and YOLO detection head [30].

The backbone is dedicated to collecting the input image and extracting feature maps from it. This is an important stage in any object detector since it is the primary structure responsible for gathering contextual information from the input image and abstracting that information into patterns. Backbones for object detectors operating on GPU platforms might be VGG, ResNet, ResNeXt, or DenseNet [30]. SqueezeNet, MobileNet, or ShuffleNet might be the backbone for object detectors running on CPU platforms [29].

The neck is important in the transmission of small-object data because it prevents information from being degraded to higher levels of abstraction [30]. It accomplishes this by upsampling the resolution of the extracted features once again, allowing distinct layers from the backbone to be consolidated and regaining influence on the detection phase. The advantage of this technique is that no feature layer aggregation procedure, such as SSD [34], is performed straight after the multi-level feature map. Path-aggregation blocks often utilized in the neck include: FPN, PANet, NAS-FPN, BiFPN, ASFF, and SFAM [30]. The use of various up-and-down sampling, splicing, dot sum, or dot product to construct aggregation algorithms is shared by all of these processes. Other blocks employed in the neck are SPP, ASPP, RFB, and CBAM [30].

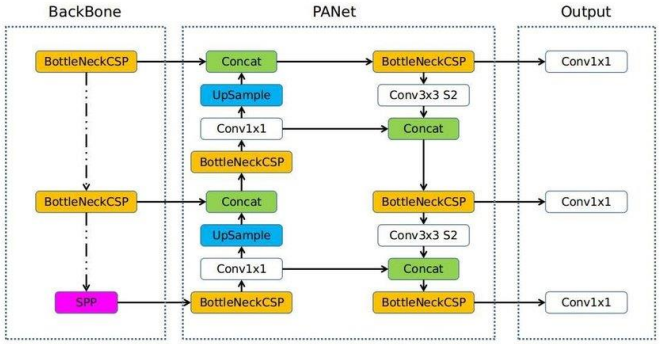

Fig. 2 Structure of YOLOv5 model [19]

The YOLOv5s backbone cannot accomplish the localization task because it is a classification network, hence the head is responsible for recognizing the location and category of the object using the features maps retrieved from the backbone. There are two types of heads: dense prediction and sparse prediction. The RCNN series is the most typical of sparse predictors, which have long been the dominating approach in the area of object detection [31]. In comparison to the sparse detector, the dense detector predicts both the bounding box and the class of objects [32]. The dense detector has a clear speed advantage, but its accuracy is poorer. The most representative models for dense detectors are the YOLO series [33].

II. Improved YOLOv5s Architectural Changes

A. Backbone

In this study, the CSPDarknet YOLOv5 backbone was replaced with DenseNet. Implementing the DenseNet structure necessitated breaking it down into its essential components and ensuring that the layers interacted properly. This involved maintaining the correct feature map size, which necessitated significantly changing the scaling factor for the width and depth of the model. It was critical to avoid dramatically changing the number of layers from the original model in order to preserve a comparable level of complexity. As a result, DenseNet was downscaled appropriately to retain its basic capability. Three convolution blocks and modules of the CSPbottleneck in the original version of YOLOv5 were also replaced with Transformer encoder blocks in order to fully utilize the global information of the bird pictures. The structure is shown in Fig. 3. Each TRANS module was divided into two sub-layers, the first of which was a multi-headed attention layer and the second of which was a completely linked layer.

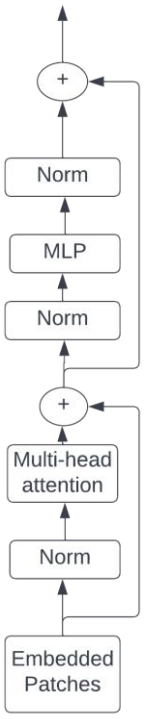

Fig. 3. The architecture of the transformer encoder

B. Neck

PANet, as shown in Fig. 2, is used in YOLOv5. However, in this paper, it was substituted with a BiFPN [34]. By adding extra weights to the input characteristics of multiple resolutions, BiFPN helps the network to comprehend the value of each feature resulting in improved integration of varied scale information. When it comes to difficult-to-identify birds, deeper features are better in abstracting the properties of the problem.

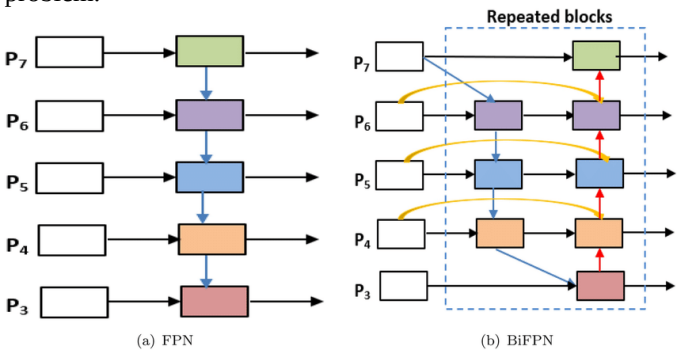

Fig. 4 Feature network design (a) PANet adds an additional bottom-up pathway on top of FPN (b) BiFPN implements two optimizations for cross-scale connections [34]

When fusing features at multiple scales, BiFPN adds weights to the input features of different resolutions rather than merely summarizing or concatenating them [35]. As illustrated in Fig. 4, if the original input nodes and output nodes are at the same level, BiFPN inserts more edges between them to fuse more information without contributing too much extra processing. BiFPN combines bidirectional cross-scale connection with trainable parameters and fast normalization fusion [36].

C. Head

In this study, one additional prediction head for tiny object detection was introduced. The four-head framework, when combined with the other three prediction heads, mitigates the detrimental impact produced by violent object size variation. The whole network structure is shown in Fig. 5.

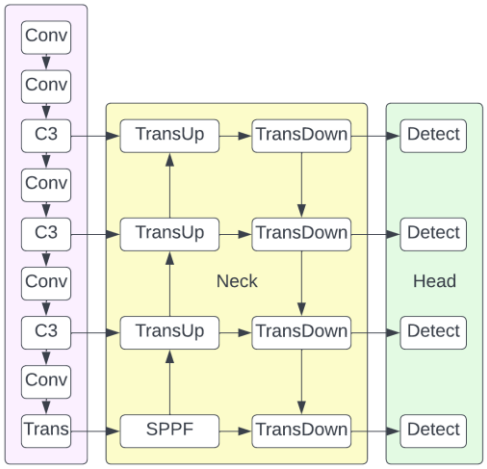

Fig. 5 The whole network structure. (1) DenseNet backbone with a TRANS module at the end. (2) Neck using the structure of the BiFPN. (3) Feature map of four detection heads using the TRANS module in neck

III. DATA PREPROCESSING AND TRAINING

a. DATASET

The dataset used in this study was the Klim dataset. The dataset contains 1607 images for training, 340 images for validation, and another 357 images for testing. Fig. 6 shows bird size statistics from the Klim datasets. Fig. 7 shows a sample image from the dataset. The mosaic data augmentation approach was utilized to enlarge the image collection before it was fed into the improved YOLOv5s network model. The photos were spliced using a variety of approaches, including random scaling, random cropping, and random arrangement, which not only increased the image collection but also enhanced the recognition of tiny objects. Furthermore, before training the model, adaptive scaling and filling operations were conducted on the training dataset, and the input image size was adjusted to 416 x 416 pixels.

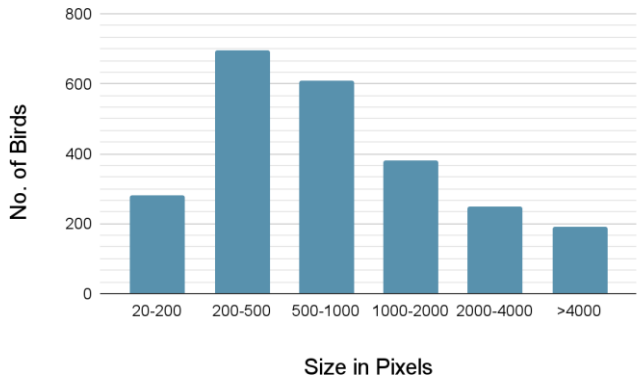

Fig. 6 Bird size statistics from Klim datasets.

b. EXPERIMENTAL EQUIPMENT

The models were implemented on Google Colab Pro cloud services with Tesla P100-PCIE 16GB GPUs.

c. MODEL TRAINING

To train the models, transfer learning was applied. Training began with YOLOv5s weights that had been learned on the MS COCO dataset. The first 10 layers of the backbone were frozen so that the weights in the backbone layers did not change during transfer learning. The Klim dataset was used to train the head layers. Grid search was used to find the best combinations for learning rates, batch size, network resolution, subdivision, and anchors. The hyperparameters were set as follows: batch size: 16, height and width: 416, sub-division: 32, momentum: 0.921, learning rate: 0.001, decay: 0.0005.

Since there is only one class, the number of filters before each of the three YOLO layers was set to 18. Stochastic Gradient Descent was used as the optimizer in the network. For a custom object detector, anchors are important parameters to tweak based on the object sizes in the annotated training dataset. In YOLO, anchor boxes are estimated using k-means clustering with cluster size of 9 on the dimensions of the ground truth bounding boxes. Each model has 9 anchor boxes to learn small, medium, and large objects. The optimized parameters for each model are listed in Table I.

TABLE I. TRAINING PARAMETERS FOR ORIGINAL YOLOV5S AND IMPROVED YOLOV5S

	Original YOLOv5	**Improved YOLOv5**
Burn in	100	100
Steps	(1600,1800)	(1400, 1700)
Scale	(0.1, 0.1)	(0.1, 0.01)
Anchors	(5,8) (7,11) (11,13) (13,18) (20,18) (20,29) (33,27) (50,47) (60,98)	(5,8) (7,11) (11,14) (15,17) (21,20) (19,31) (32,27) (48,46) (61,97)

The model was then trained for 100 epochs. Fig. 8 shows the flow diagram of the training process. The best network weight was acquired when the training was done. Following that, the improved YOLOv5s performance was assessed using the test set and compared to the test results of the original YOLOv5 network.

Fig. 7 Sample image from the Klim dataset

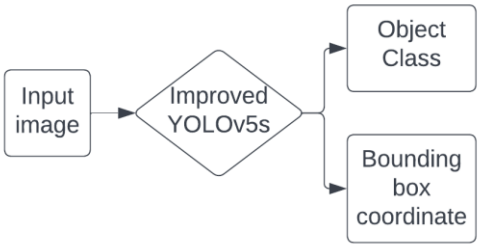

Fig. 8 The flow diagram of the training method

VI. EXPERIMENTAL RESULTS

CONVERGENCE RESULTS OF THE NETWORK MODEL

The training and validation sets were inputted into the network for training. Both the original YOLOv5s and improved YOLOv5s models were trained using images from the Klim dataset. The dataset includes 1607 objects for training, 340 for validation, and another 357 for testing. After 100 epochs of training, the loss function value curves of the training and verification sets for the improved YOLOv5s were determined as shown in Fig. 9, and they included the detection frame loss, the detection object loss, and the class loss. The graphs show a good fit between validation and training data, confirming that the model was not suffering from over-fitting or under-fitting. The detection frame loss reflects if an algorithm can accurately locate the center point of an object and whether the detection target is covered by the projected bounding box.

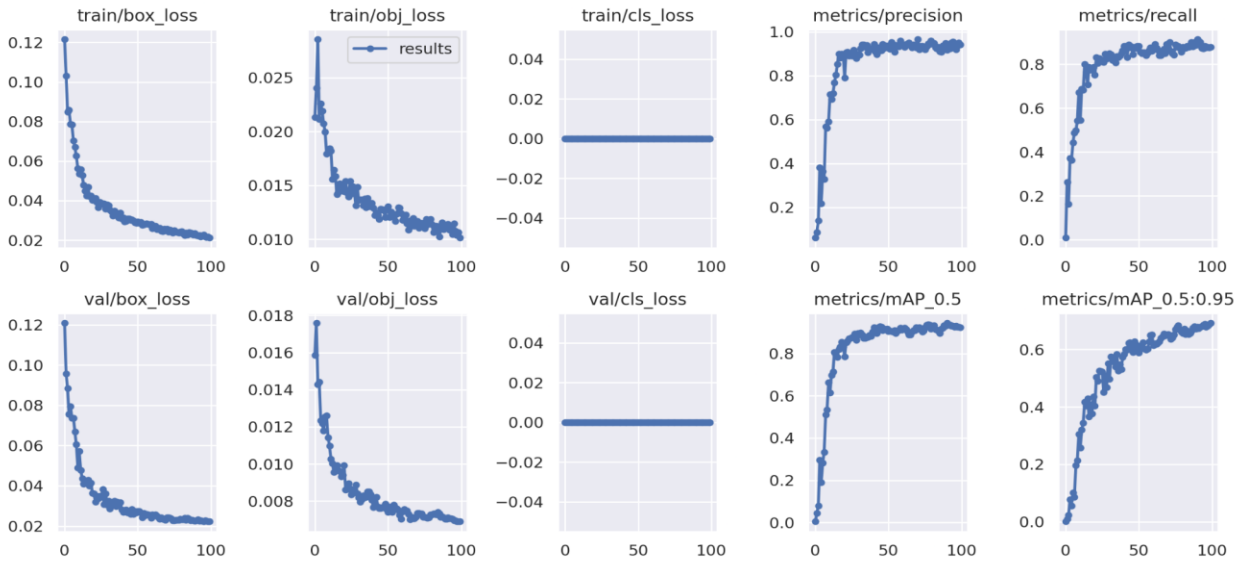

Fig. 9 Training and validation curves for improved YOLOv5s for the 100 training epochs

The more precise the prediction frame, the less the loss function value. The object loss function is simply a probability measure of the presence of the detection target in the region of interest. The better the precision, the smaller the value of the loss function. The class loss measures the ability of the algorithm to properly anticipate a certain item category. The lower the loss value, the better the categorization. As shown in Fig. 9, the loss function value had a downward trend during the training process, the Stochastic Gradient Descent algorithm optimized the network and the network weight and other parameters were constantly updated. Before the training batch reached 20, the loss function value dropped rapidly, and the accuracy, recall rate and average accuracy rapidly improved. The network continued to iterate.

When the training epochs reached approximately 50, the decrease in the loss function value gradually slowed. Similarly, the increases in parameters such as average accuracy also slowed. When the training epochs reached 50 the loss curves of the training and validation sets showed almost no downward trends, and other index values also tended to have stabilized. The network model basically reached the convergence state, and the optimal network weight was obtained at the end of training.

VERIFICATION OF THE NETWORK MODEL

It was critical to utilize proper assessment measures for each problem while evaluating the detection performance of both the original YOLOv5s and improved YOLOv5s network. The evaluation measures were precision, recall, average precision, F1 score, and mean average precision, which were defined as follows:

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP+FN} \quad (2)$$

$$AP_i = \int_0^1 P(R)d(R) \quad (3)$$

$$F1 = \frac{2PR}{P+R} \quad (4)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (5)$$

Where: True positives (TP) indicate that birds have been recognized and that there are birds in the real image. True negatives (TN) indicate that no birds were identified and that no birds were present in the real image. False positives (FP) are recognized when there is no bird in the picture. FN (false negatives) indicates that no birds were identified, yet there were birds in the real image.

The recall rate is used to calculate the proportion of birds identified to the total bird price in the sample. The accuracy rate is used to calculate the percentage of accurate birds detected out of all birds detected. When the two are close, F1 score is used. The greater the F1 score, the better the algorithm. In all, 357 images from the Klim dataset were utilized as the test set, and the test results for both the original YOLOv5 network and the improved YOLOv5 network included recall rate (R), accuracy

rate (P), and mAP score. Table II shows the results of the original YOLOv5s and improved YOLOv5s models evaluated on the Klim dataset.

As the data in Table II shows, original YOLOv5s has the fastest detection speed but the lowest mAP. Fig. 10 shows that the original YOLOv5s has a lower detection accuracy for small birds compared to the improved YOLOv5s model. The proposed method based on YOLOv5s has better detection accuracy in both relatively static and highly dynamic backgrounds images in the Klim dataset compared to the original YOLOv5s as shown in Fig. 11 where the improved YOLOv5s accurately localized the bird and Fig. 12 where the original YOLOv5s was unable to localize the bird. The original YOLOv5s model achieved a mAP score of 84.1% on the validation set. The detection threshold was set at 0.5. Improved YOLOv5s model achieved a mAP score of 88.1% on the test data. This was a 4.8% improvement over the original YOLOv5s model.

TABLE II. AP_{50} AND MAP SCORES FOR ORIGINAL YOLOV5S AND IMPROVED YOLOV5S MODELS ON THE KLIM DATASET

	Original YOLOv5s	**Improved YOLOv5s**
Training/Validation/Test Set	1607/340/357	1607/340/357
AP50 (%) (Validation)	82.5	87.3
mAP(%) (Validation)	81.7	84.1
AP50 (%) (Test)	82.5	83.2
mAP(%) (Test)	83.3	88.1

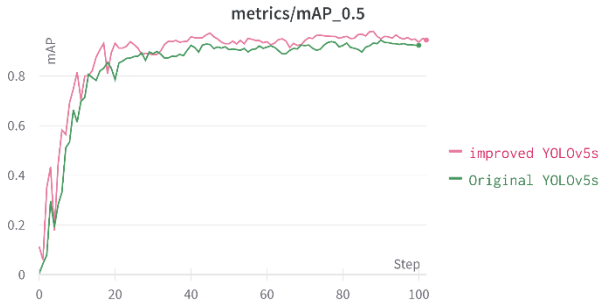

Fig. 10 mAP curves of improved YOLOv5s and original YOLOv5

Fig. 11 Visualization results from our Improved YOLOv5s on Klim dataset

Fig. 12 Visualization results from the original YOLOv5s on Klim dataset

VII. CONCLUSION

This paper identified architectural modifications to YOLOv5s that deliver a clear improvement in performance at relatively low computational cost, as the improved YOLOv5s model retains real-time inference speed and better detects smaller objects. The scenario in which the proposed approach is used, that of autonomous bird deterrent, is one that would benefit substantially from such an upgrade. Comparing Fig. 11 and Fig. 12, such modifications have a quantifiable effect on bird detection on the Klim dataset. The proposed approach only significantly enhanced the performance of the baseline model, but also identified a number of specific strategies that may be used for any other application requiring the detection of small distant objects.

Finally, while the empirical advantages of the proposed architectural improvements are significant in this study, the consistency and applicability of the results should be studied further. For example, the analysis might benefit substantially from more testing with diverse datasets. This would be a significant step toward a more robust small object detection model.

ACKNOWLEDGMENT

The authors also acknowledge the financial support of the AFRICA-*ai*-JAPAN Project (JICA).

REFERENCES

[1] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-

offs for modern convolutional object detectors," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2]Hossain and Lee, "Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices," *Sensors*, vol. 19, no. 15, p. 3371, 2019.

[3]A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. de la Iglesia, I. González, J.-P. Mercier, G. Gagné, A. Mitra, and S. Rajashekar, "Drone vs. bird detection: Deep learning algorithms and results from a Grand Challenge," *Sensors*, vol. 21, no. 8, p. 2824, 2021.

[4]L. B. Boudaoud, F. Maussang, R. Garello, and A. Chevallier, "Marine Bird detection based on deep learning using high-resolution aerial images," *OCEANS 2019 - Marseille*, 2019.

[5]K. Zhang, Y. Musha, and B. Si, "A rich feature fusion single-stage object detector," *IEEE Access*, vol. 8, pp. 204352–204359, 2020.

[6]D. Li, D. Zhao, Y. Chen, and Q. Zhang, "DeepSign: Deep Learning Based Traffic Sign Recognition," *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018.

[7]E. Vasilopoulos, G. Vosinakis, M. Krommyda, L. Karagiannidis, E. Ouzounoglou, and A. Amditis, "A comparative study of autonomous object detection algorithms in the maritime environment using a UAV platform," *Computation*, vol. 10, no. 3, p. 42, 2022.

[8]Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using Yolo-V5 algorithm for fruit automatic loading system," *Postharvest Biology and Technology*, vol. 185, p. 111808, 2022.

[9]J. Fang, Q. Liu, and J. Li, "A deployment scheme of Yolov5 with inference optimizations based on the Triton Inference Server," *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 2021.

[10]I. S. Isa, M. S. Rosli, U. K. Yusof, M. I. Maruzuki, and S. N. Sulaiman, "Optimizing the hyperparameter tuning of yolov5 for underwater detection," *IEEE Access*, vol. 10, pp. 52818–52831, 2022.

[11]C. Devaguptapu, N. Akolekar, M. M. Sharma, and V. N. Balasubramanian, "Borrow from anywhere: Pseudo Multi-Modal Object Detection in thermal imagery," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[12]Y. Mei, K. Wu, Z. Xu, H. Shan, and M. Wang, "SNG-Yolox: Non-obvious remote sensing target detection based on Enhanced Yolox," 2022.

[13]Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[14] P. Pham, D. Nguyen, T. Do, T. Duc, and D.-D. Le, *Evaluation of Deep Models for Real-Time Small Object Detection*. 2017, p. 526. doi: 10.1007/978-3-319-70090-8_53.

[15]C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16]M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17]A. Ahmed, P. Tangri, A. Panda, D. Ramani, and S. Karmakar, "Vfnet: A convolutional architecture for accent classification," in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4.

[18]J. Wang, Y. Chen, M. Gao, and Z. Dong, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," *arXiv preprint arXiv:2112.08782*, 2021.

[19] Z. Qu, L.-yuan Gao, S.-ye Wang, H.-nan Yin, and T.-ming Yi, "An improved YOLOV5 method for large objects detection with multi-scale feature cross-layer fusion network," *Image and Vision Computing*, vol. 125, p. 104518, 2022.

[20] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[21] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "SCRDET: Towards more robust detection for small, cluttered and rotated objects," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[22] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network," *Remote Sensing*, vol. 12, no. 9, p. 1432, 2020.

[23]F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, and F. Herrera, "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance," *Knowledge-Based Systems*, vol. 194, p. 105590, 2020.

[24]Z. Liu, G. Gao, L. Sun, and Z. Fang, "HRDNet: High-resolution detection network for small objects," *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021.

[25]P. Soviany and R. T. Ionescu, "Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction," *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2018.

[26]Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications*, vol. 172, p. 114602, 2021.

[27]P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-Tiny: Object Detection and recognition using one stage improved model," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020.

[28]K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on Deep Learning: A Review," *Image and Vision Computing*, vol. 97, p. 103910, 2020.

[29]"Table 1: Yolo object detection comparison between Yolov4 ONNX, Yolov4 Darknet, Yolov4 Darknet tiny Yolov4, pp-yolo, opencv leaky Yolov4 and opencv yolov4."

[30] L. Li, Z. Shuai, J. Hu, and Y. Zhang, "Classification of tropical cyclone intensity based on deep learning and Yolo V5," *Advances in Artificial Intelligence and Security*, pp. 280–291, 2022.

[31]Y. Long, D. Jin, Z. Wu, Z. Zuo, Y. Wang, and Z. Kang, "Accurate identification of infrared ship in island-shore background based on visual attention," *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2022.

[32]P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "Sparse R-CNN: End-to-end object detection with learnable proposals," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[33]X. Zhang, W. Yang, X. Tang, and J. Liu, "A fast learning method for accurate and robust lane detection using two-stage feature extraction with Yolo V3," *Sensors*, vol. 18, no. 12, p. 4308, 2018.

[34]W. Fang, L. Wang, and P. Ren, "Tinier-yolo: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020.

[35]J. Chen, H. S. Mai, L. Luo, X. Chen, and K. Wu, "Effective feature fusion network in BIFPN for small object detection," *2021 IEEE International Conference on Image Processing (ICIP)*, 2021.

[36] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, "Vit-Yolo:transformer-based Yolo for object detection," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.