

Enhancing Long-Tailed Macaque Pest Identification Using a Two-Stage YOLO-EfficientNet Pipeline

Hammami Rifqi Darmawan

Department of Informatics

Universitas Muhammadiyah Banjarmasin

Banjarmasin, Kalimantan Selatan

hammami_rifqi_2255201110035@umbjm.ac.id

Finki Dona Marleny

Department of Informatics

Universitas Muhammadiyah Banjarmasin

Banjarmasin, Kalimantan Selatan

finkidona@umbjm.ac.id

Ayu Ahadi Ningrum

Department of Informatics

Universitas Muhammadiyah Banjarmasin

Banjarmasin, Kalimantan Selatan

ayuahadi@umbjm.ac.id

Abstract—Long-tailed macaque (*Macaca fascicularis*) infestations have reduced corn production from a potential 7–9 tons to only 4–5 tons per growing season ($\pm 40\%$), while conventional deterrence methods such as loud noise and shooting have proven ineffective. This study developed an Internet of Things (IoT)-based pest repellent system using audiosonic waves, an HC-SR501 sensor, and an ESP32-CAM module, incorporating YOLOv8 and EfficientNet-B0 to replace the previously used YOLOv3 Tiny model, with the aim of improving detection accuracy and distinguishing long-tailed macaques from humans. Since population density at the partner's farmland was insufficient to construct an adequate dataset, image data were collected at Kembang Island, South Kalimantan, which serves as a natural habitat supporting a large and observable population of long-tailed macaques, thereby enabling the acquisition of visually representative and diverse training data. A total of 3,083 images were annotated using Label Studio under two classes (human and long-tailed macaque), then divided at an 80:10:10 ratio without augmentation. The YOLOv8n model was trained for 50 epochs, after which validation detections were cropped into 920 images to fine-tune EfficientNet-B0 over 10 epochs. Evaluation results show that YOLOv8 achieved a precision of 0.959, recall of 0.965, mAP50 of 0.979, and mAP50-95 of 0.723, while EfficientNet-B0 reached validation accuracies of 98.4–100% with no significant overfitting. The two-stage detection-classification pipeline was successfully tested end-to-end on 121 field images, demonstrating substantial performance improvements over the prior model and contributing toward IoT and AI-based smart agriculture solutions for effective, sustainable corn field protection.

Keywords—IoT, YOLOv8, EfficientNet, long-tailed macaque, pest repellent, smart agriculture.

I. INTRODUCTION

Long-tailed macaques (*Macaca fascicularis*) are a primate species with a wide distribution across Indonesia and are often classified as harmful agricultural pests. This species' high adaptability to various habitat types, including agricultural areas bordering forest zones, causes groups of long-tailed macaques to frequently enter and damage community-owned farmland [1]. Numerous cases of human-long-tailed macaque conflicts have been reported across various regions in Indonesia, where attacks by these primate groups have caused damage to a variety of food crops, including corn, cassava, and vegetables. The resulting crop losses are significant, while farmers' tolerance for the presence of this pest is relatively low [2].

This problem is also directly experienced by the corn farmers who are partners in this study. The land they

manage actually has a potential yield ranging from 7 to 9 tons per growing season. However, due to repeated attacks by long-tailed macaques, the actual harvest yields only range from 4 to 5 tons, representing a decrease of approximately 40% from the potential that could have been achieved. Several conventional control efforts have been attempted, including scaring them away with loud noises or shooting, but these methods have not been able to provide effective and sustainable results. This situation aligns with the findings of a study by Fitria *et al* [1], which noted that conventional short-term control methods are generally insufficient to prevent primates from re-entering and attacking agricultural land.

Advances in Internet of Things (IoT) technology offer significant opportunities to introduce more innovative solutions for pest control in the agricultural sector. IoT-based systems that integrate motion sensors, microcontrollers, and automatic deterrent mechanisms have proven capable of significantly enhancing the effectiveness of land protection. One approach considered quite promising is the use of audiosonic waves as a deterrent, which can be automatically activated when the system detects the presence of an object in the vicinity of the field [3], [4]. In this system, the HC-SR501 Passive Infrared (PIR) sensor functions as an object detector based on emitted infrared radiation, while the ESP32-CAM module serves as a real-time visual image capture unit.

In previous research, the author designed an IoT-based pest repellent prototype equipped with the YOLOv3 Tiny object detection model to automatically recognize the presence of long-tailed macaques. Although the results obtained showed considerable potential, the YOLOv3 Tiny model still has several shortcomings, particularly regarding detection accuracy and its ability to distinguish pest objects from non-pest objects, such as humans. These shortcomings risk causing detection errors in the form of false positives, which in turn can disrupt human activities around the land.

To address these shortcomings, this study adopts the YOLOv8 architecture as the primary detection model. Relative to its predecessors, YOLOv8 introduces several architectural advancements, including an anchor-free detection mechanism, mosaic data augmentation, and an optimized convolutional structure through the C2f module, collectively improving detection performance while reducing inference latency. Empirical evidence from multiple plant pest detection studies has further confirmed that YOLOv8 consistently outperforms earlier YOLO generations under complex environmental conditions [5], [6].

To further strengthen classification capability, this study integrates EfficientNet-B0 as a secondary classifier alongside YOLOv8. While YOLOv8 excels at rapid, lightweight object localization, its classification output on visually similar classes, such as long-tailed macaques and humans, may remain susceptible to error when operating under constrained computational environments. EfficientNet-B0 addresses this gap through its compound scaling mechanism, which proportionally scales network depth, width, and resolution to achieve high classification accuracy with minimal computational overhead. The combination of YOLO-based detection and EfficientNet-based classification has demonstrated strong performance across agricultural visual recognition tasks, including plant growth status identification in hydroponic systems [7] and pest detection and classification in crop monitoring applications, as demonstrated in studies by Garcia *et al* [5] and Raza *et al* [6]. Rather than relying on a single model to simultaneously handle both localization and classification, a design that imposes greater computational demand and reduces per-task optimization, this study deliberately separates these responsibilities across two purpose-optimized architectures. This task separation strategy has been shown to yield higher overall system accuracy while remaining tractable on edge devices with limited resources. Additionally, to support the evaluation of the performance of the detection and segmentation models used, this study also refers to several image evaluation metrics commonly used in digital image processing studies [8].

The integration of YOLOv8 and EfficientNet-B0 within this audiosonic IoT-based pest repellent system leverages the comparative strengths of each architecture: YOLOv8n provides fast, low-latency object detection in real time, while EfficientNet-B0 refines classification outcomes on cropped detection regions with greater precision, thereby substantially reducing false positive errors even under the computational constraints of an edge device. The developed system is thus expected not only to deliver more effective and sustainable protection for the partner's corn fields, but also to contribute a replicable framework for smart agriculture solutions that harness the combined potential of artificial intelligence and IoT technology in addressing real-world agricultural challenges.

II. METHODS

A. Data Collection

The first stage of this study was the collection of imagery to be used as training data for the object detection model. Data collection was conducted through direct field surveys on Kembang Island, South Kalimantan, a conservation area known as the natural habitat of a relatively large population of long-tailed macaques (*Macaca fascicularis*). This location was selected based on the consideration that the visual characteristics of individuals in this population, including body morphology, fur color, posture, and movement behavior, bear a high resemblance to the long-tailed macaque populations that act as pests on farmers agricultural lands. Thus, the collected data is expected to represent the

characteristics of the real-world objects that the system will detect in the field, as illustrated in Figure 1.



Figure 1. Image acquisition on Kembang Island, South Kalimantan

In addition to long-tailed macaques, some images also include humans (visitors or researchers) within the frame. These images containing human subjects were intentionally retained and will later be used as a comparison class (humans) during the annotation phase, with the aim of enabling the model to learn the visual differences between the two classes from the early stages of training. This data collection approach, which incorporates variations in distance, viewpoint, lighting, and the number of objects, aligns with common practices in YOLOv8-based wildlife detection research, which emphasizes the importance of diverse image acquisition conditions to enhance the model's generalization ability in real-world scenarios [9]. Through this entire data collection process, a total of 3,083 images were successfully collected, which were subsequently used as the main dataset during the annotation phase.

B. Device Specifications

All model training processes in this study were conducted using Google Colab, a cloud-based computational platform that provides free access to accelerated hardware resources without requiring local high-performance infrastructure. Training was performed on an NVIDIA Tesla T4 GPU, which offers 16 GB of GDDR6 memory and is well-suited for deep learning workloads involving both object detection and image classification tasks. The use of a cloud-based GPU environment was particularly advantageous in this study, given that local hardware available on the Windows 10 development machine was insufficient to support the computational demands of training large-scale deep learning models within a reasonable timeframe.

On the software side, the entire training pipeline was developed and executed within Jupyter-based Google Colab Notebooks using Python 3.10 as the primary programming language. Object detection model training was carried out using the Ultralytics YOLOv8 framework, which provides a high-level API for model configuration, training, validation, and inference. Image classification training for the EfficientNet-B0 model was implemented using PyTorch, leveraging its pretrained model repository via the torchvision.models module for weight initialization prior to fine-tuning. Supporting libraries including OpenCV, NumPy, and Matplotlib were additionally

employed for image preprocessing, numerical computation, and training visualization, respectively.

C. Data Annotation

The next stage involved annotating all 3,083 collected images. The annotation process was conducted using the Label Studio application [10], an open-source data annotation software commonly used to prepare datasets for computer vision tasks. In this stage, every object present in the image was marked using a bounding box and categorized into one of two label classes: human and long-tailed macaque, as shown in Figure 2.



Figure 2. Image annotation in the Label Studio application

After all images were annotated, the annotation results were exported into the YOLO dataset format. This format consists of two main components: image files and label files containing bounding box coordinates along with class indices for each object in each image, in accordance with the directory structure required by the Ultralytics YOLOv8 framework [11].

D. Dataset Split

The annotated dataset, which has been exported to the YOLO format, is then divided into three subsets: 80% training data, 10% test data, and 10% validation data. This division aims to ensure the model receives sufficient training data to learn the visual patterns of both object classes, while also having independent validation and test subsets to objectively evaluate the model's performance on previously unseen data. This 80:10:10 split ratio is a commonly used and recommended ratio in the YOLOv8 model training process on custom datasets, based on research by Nagaraj *et al* [9].

At this stage, the researchers intentionally did not apply additional data augmentation techniques to the YOLOv8 dataset. This decision was made with the consideration that excessive augmentation, particularly on images with complex natural backgrounds, has the potential to introduce artificial visual variations that could increase the risk of false positive detection errors when the model is implemented in real-time in the field; thus, the amount of original data was deemed representative for the initial training stage.

E. YOLOv8 Model Configuration and Training

The next step is to train the object detection model using the YOLOv8 architecture. The dataset configuration is specified in the data.yaml file, which contains information on the images and labels directories resulting from the data split, the number of classes (nc) (2), and the class names (names): human and long-tailed macaque.

The training process is performed by loading the pre-trained YOLOv8n model as the base model, followed by fine-tuning with the following configuration:

- Number of *epochs*: 50
- Input image *size*: 640×640 pixels
- *Batch size*: 16

The selection of the YOLOv8n (nano) variant was based on computational efficiency considerations, given that this model will eventually be implemented on an edge device, specifically an ESP32-CAM, which has limited resources. YOLOv8 itself introduces several architectural updates compared to previous YOLO generations, including an anchor-free detection mechanism with decoupled heads, as well as the C2f module (Cross-Stage Partial bottleneck with two convolutions) in the backbone and neck sections, which collectively improve detection accuracy while accelerating the inference process [12]. The input image resolution of 640×640 pixels was chosen because it is a commonly used standard size and has been shown to provide a good balance between detection accuracy and computational efficiency in the research presented by Nagaraj *et al* [9], on YOLOv8-based wildlife detection.

F. EfficientNet Dataset Preparation

After the YOLOv8 model training process was completed, the model with the best performance in the 'best.pt' file was used to crop the detected objects in the validation dataset subset with a confidence threshold of 0.5, so that only objects detected with a confidence level above 50% would be cropped and saved. The cropping results, consisting of individual object images for each class (humans and long-tailed macaques), are then organized into a new dataset to be used in the training phase of the EfficientNet classification model. This two-step pipeline, consisting of object detection followed by classification of the cropped images, has proven effective in improving object recognition accuracy across various deep learning application studies, where the output of the detection model serves as input for the classification model in the subsequent stage [13].

G. EfficientNet-B0 Model Configuration and Training

In this stage, the classification model is configured using the EfficientNet-B0 architecture, with an input image resolution of 224×224 pixels, in accordance with the standard resolution recommended for this architecture. The cropped dataset from the previous stage is loaded using a class-based directory structure and automatically split into 80% training data and 20% validation data. In this stage, simple data augmentation in the form of random horizontal flipping is applied to the training data, along with image normalization using the mean and standard deviation of the ImageNet dataset, in accordance with the standard configuration for pre-trained EfficientNet models [14].

Next, a data loader was created for each data subset with a batch size of 16. The pre-trained EfficientNet-B0 model was loaded as the base model, and the final

classification layer in that architecture was modified to produce two output classes: humans and long-tailed macaques. The loss function was configured using Cross Entropy Loss, while the optimizer used Adam with a learning rate of 0.0005, a relatively small value intended to fine-tune the weights of the pre-trained model without damaging the features previously learned on the ImageNet dataset. The training of the EfficientNet-B0 model was conducted over 10 epochs and was successfully completed without technical issues.

III. RESULTS AND DISCUSSION

A. YOLOv8 Model Training Results

The evaluation results of the YOLOv8 model after 50 training epochs on the validation data are presented in Table 1 below.

Table 1. YOLOv8 Model Evaluation Results on Validation Data

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	308	909	0.959	0.965	0.979	0.723
human	25	63	0.949	0.952	0.974	0.722
monyet ekor panjang	307	846	0.970	0.978	0.985	0.724

As presented in Table 1, the fine-tuned YOLOv8n model attains strong overall detection performance across both classes, reflected by an aggregate precision of 0.959, recall of 0.965, mAP50 of 0.979, and mAP50-95 of 0.723. The near-perfect mAP50 score suggests that the model reliably localizes objects with high spatial correctness at an Intersection over Union (IoU) threshold of 0.5, which is the standard benchmark for object detection evaluation.

At the class level, the long-tailed macaque class yields the highest individual scores, achieving a precision of 0.970, recall of 0.978, and mAP50 of 0.985. This outcome is attributable to the substantially greater number of annotated instances available for this class (846 instances) relative to the human class (63 instances), providing the model with richer intra-class variation from which to extract discriminative visual features. Despite the pronounced class imbalance, the human class nonetheless achieves a precision of 0.949 and a recall of 0.952, demonstrating that the model retains adequate discriminative capacity even for the minority class. This resilience may be partly explained by the visually distinct morphological characteristics between the two classes, differences in body structure, limb proportion, and movement posture, which offer sufficiently separable feature representations even with limited training samples.

A notable disparity exists between the mAP50 (0.979) and mAP50-95 (0.723) scores, revealing that while the model performs robustly under lenient IoU criteria, its bounding box localization precision diminishes as stricter

overlap requirements are imposed. This behavior is characteristic of lightweight detection architectures such as YOLOv8n, wherein the reduced parameter count and simplified feature extraction hierarchy, designed to minimize inference latency on constrained hardware like the ESP32-CAM, inherently limit the model's ability to produce tightly fitted bounding boxes, particularly around objects with irregular contours or partially occluded body regions. Despite this trade-off, the mAP50 score of 0.979 surpasses the performance range of 0.89–0.94 reported in comparable wildlife detection studies by Terven et al. [12] and Tan and Le [14], which used YOLOv8, thereby confirming the effectiveness of collecting domain-specific training data directly from the natural habitats of the target species.

Figure 3 shows the training and validation curves for the three loss metrics (box loss, classification loss, and distribution focal loss/DFL) over 50 epochs. All three loss curves, for both the training and validation data, exhibit a consistent downward trend with no significant signs of overfitting, as the validation loss curve continues to follow the downward trend of the training loss curve until the final epoch. The box loss value decreased from 1.375 in the first epoch to 0.813 in the 50th epoch, while the classification loss decreased sharply from 1.649 to 0.361. This decrease indicates that the model is getting better at learning both the location and classification of objects as the number of epochs increases.

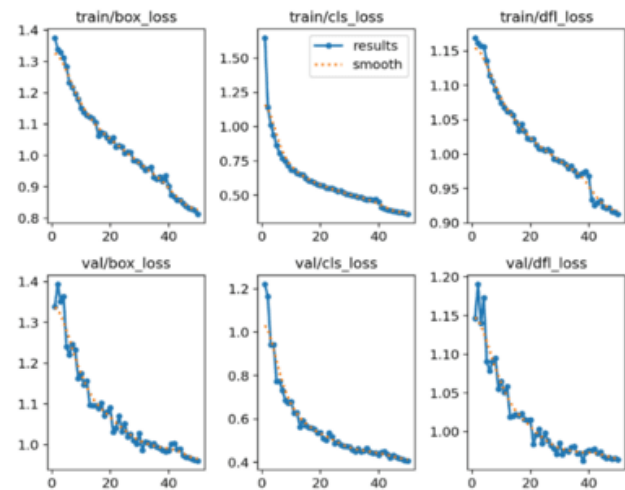


Figure 3. Box Loss, Classification Loss, and DFL Loss curves on training and validation data

Figure 4 presents the evolution of four detection metrics, precision, recall, mAP50, and mAP50-95, across the 50-epoch training run. All four metrics exhibit a consistent upward trajectory, with mAP50 surpassing the 0.95 threshold from approximately epoch 25 onward and ultimately reaching 0.981 at the final epoch. The stable plateauing behavior observed in later epochs confirms that the chosen training duration of 50 epochs was adequate for the model to reach satisfactory convergence, with no indication that additional epochs would yield meaningful performance gains.

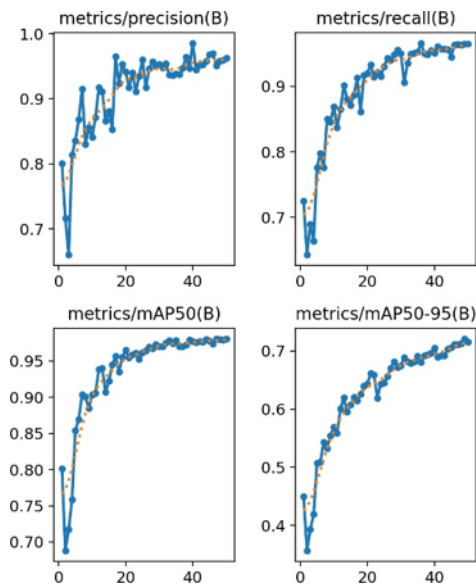


Figure 4. Precision, Recall, mAP50, and mAP50-95 curves during training

When contextualized against prior work employing YOLOv3 Tiny, the improvements recorded here are substantial across both classification accuracy and bounding box localization quality, reinforcing findings from multiple studies by Garcia *et al.* [5] and Raza *et al* [6], which position YOLOv8 as a consistently superior architecture relative to its predecessors in agricultural and wildlife detection tasks. Furthermore, the mAP50 of 0.979 achieved in this study exceeds the 0.89-0.94 range reported in comparable YOLOv8-based wildlife detection studies by Nagaraj *et al* [9] and Zhu *et al* [15]. This elevated performance is largely attributable to two dataset-level factors: the taxonomic homogeneity of the target classes, one animal species paired with a single non-target class, which yields well-separated inter-class feature distributions, and the ecological alignment between the data collection site (Kembang Island) and the intended deployment environment, which ensures that the learned visual representations closely reflect conditions encountered during actual field operation.

B. EfficientNet-B0 Model Training Results

The best.pt model resulting from YOLOv8 training was subsequently used to crop objects in the validation dataset subset with a confidence threshold of 0.5. This process yielded a total of 920 cropped images divided into two classes: humans and long-tailed macaques. This set of cropped images was then automatically split into 80% training data (736 images) and 20% validation data (184 images) for training the EfficientNet-B0 classification model.

The EfficientNet-B0 fine-tuning process was performed by replacing the final classification layer with a 2-class output, using the Adam optimizer with a learning rate of 0.0005 and the Cross Entropy loss function. Training was conducted for 10 epochs, with the results presented in Table 2.

Table 2. EfficientNet-B0 Model Training Results Over 10 Epochs

Epoch	Train Loss	Train Acc (%)	Val Loss	Val Acc (%)
1	0.1179	96.9	0.0129	98.9
2	0.0216	99.5	0.0038	100.0
3	0.0168	99.5	0.0040	100.0
4	0.0016	100.0	0.0024	100.0
5	0.0126	99.5	0.0805	98.4
6	0.0045	100.0	0.0333	99.5
7	0.0133	99.3	0.0180	99.5
8	0.0078	99.7	0.0057	99.5
9	0.0290	99.0	0.0294	99.5
10	0.0158	99.6	0.0050	100.0

As shown in Table 2, the fine-tuned EfficientNet-B0 model exhibits remarkably rapid convergence throughout the training process. Training and validation accuracies both exceeded 96% as early as the first epoch, subsequently stabilizing within the 99–100% range across the remaining epochs. This swift convergence can be attributed to several compounding factors. First, the input images fed to EfficientNet-B0 were pre-cropped object regions produced by the YOLOv8 detector, meaning that background clutter and irrelevant spatial context had already been eliminated prior to classification. This substantially reduces the feature-learning burden on the classifier, allowing it to focus exclusively on class-discriminative visual attributes. Second, the pronounced morphological differences between the two target classes, encompassing variations in body silhouette, limb geometry, facial structure, and surface texture, provide highly separable feature distributions that align well with the pretrained ImageNet representations embedded within the EfficientNet-B0 backbone, further accelerating convergence during fine-tuning.

Transient spikes in validation loss are observable at epoch 5 and epoch 9, both of which recover promptly in subsequent epochs without a corresponding decline in validation accuracy. These fluctuations are consistent with the stochastic behavior commonly encountered during fine-tuning on small validation sets, in this case comprising only 184 images, where individual misclassifications carry disproportionate weight in the aggregate loss computation. Importantly, the absence of a sustained upward trend in validation loss, combined with persistently high validation accuracy throughout all ten epochs, confirms that the model generalizes effectively to unseen samples without exhibiting overfitting. These results collectively demonstrate that the two-stage pipeline architecture, wherein object detection and classification are delegated to purpose-optimized models, produces a system that is both highly accurate and computationally efficient within the constraints of an edge deployment environment.

Figure 5 presents the loss and accuracy trajectories recorded during the EfficientNet-B0 fine-tuning process across 10 epochs. Transient elevations in validation loss are observable at epoch 5 and epoch 9, both of which recover in the immediately following epochs without inducing a corresponding decline in validation accuracy. These intermittent spikes are consistent with the stochastic

variance inherent to fine-tuning procedures conducted on small validation partitions, in this instance comprising 184 images, wherein isolated misclassification events exert disproportionate influence on the aggregate loss value. Crucially, neither spike precipitates a sustained divergence between training and validation loss, and validation accuracy remains anchored within the 98.4-100% band throughout all epochs, confirming that the model generalizes reliably to unseen samples.

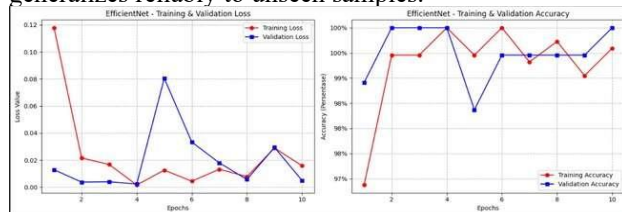


Figure 5. Training/Validation Loss and Accuracy Curves for EfficientNet-B0

The elevated and stable classification accuracy observed across epochs further substantiates the efficacy of the two-stage pipeline design, wherein YOLOv8 assumes responsibility for object localization while EfficientNet-B0 independently handles the classification decision on pre-cropped regions. This architectural separation reduces the per-model optimization burden and has been demonstrated to yield higher aggregate system accuracy in deep learning pipelines that involve visually similar target classes. Within this study's context, the secondary classification stage introduced by EfficientNet-B0 is anticipated to materially strengthen the system's discriminative capacity between long-tailed macaques and humans, thereby mitigating the false positive vulnerability that constituted the principal failure mode of the prior YOLOv3 Tiny-based system.

C. Implementation of a Two-Stage Pipeline on Field Test Data

As a final step, the pre-trained YOLOv8 and EfficientNet-B0 models were integrated into a two-stage pipeline to be tested on a new field image dataset consisting of 121 images in various file formats. In this pipeline, each input image is first processed by the YOLOv8 model (Stage 1) with a confidence threshold of 0.4 to detect and localize objects. Each detected object is then cropped directly in memory and passed to the EfficientNet-B0 model (Stage 2) for reclassification to determine the final object label (human or long-tailed macaque) along with its confidence score.

The final output of this pipeline consists of images annotated with visual bounding boxes using color-coded labels, green for the human class and red for the long-tailed macaque class, as shown in Figure 6, along with the labels and confidence scores from the EfficientNet-B0 classification. If no objects are detected in an image, the image is saved in its original state to indicate that the area within the image is safe (no humans or long-tailed macaques are present).



Figure 6. Testing with the YOLOv8-EfficientNet-B0 pipeline

IV. CONCLUSION

This study established an IoT-based audiosonic pest deterrent system for long-tailed macaque (*Macaca fascicularis*) intrusion in corn farming by advancing from a single-model YOLOv3 Tiny architecture to a two-stage pipeline combining YOLOv8n for real-time object detection and EfficientNet-B0 for secondary classification. The proposed pipeline demonstrated marked improvements in detection accuracy and inter-class discriminability, effectively reducing false positive errors that undermined the reliability of the preceding system. These outcomes confirm that architecturally separating detection and classification responsibilities across two purpose-optimized models produces a system that is simultaneously accurate and computationally viable on resource-constrained edge devices, a contribution with broad replicability for IoT and AI-based wildlife pest management in smart agriculture contexts.

Nonetheless, the system has not yet been validated under authentic operational conditions on an ESP32-CAM unit in actual cornfield environments, and the dataset's geographic exclusivity to Kembang Island constrains the verified generalizability of the trained models. Future work should therefore prioritize direct on-device deployment trials to quantify real-world inference latency, while expanding the training dataset across varied agricultural environments, lighting conditions, and macaque population densities to enhance model robustness and broaden the system's applicability to diverse pest management scenarios.

REFERENCES

- [1] W. Fitria, A. Nur Bambang, and J. Wasiq Hidayat, "Human and long-tailed macaque conflict in Central Java, Indonesia," *E3S Web of Conferences*, vol. 202, p. 06011, Nov. 2020, doi: 10.1051/e3sconf/202020206011.
- [2] K. Ilham, "Monkeys Feeding in Farm: Assessing Patterns of Crop-Feeding Long-Tailed Macaque (*Macaca fascicularis*) in West Sumatra, Indonesia and its Management Conflicts," The Rufford Foundation, 2019. [Online]. Available: <https://www.rufford.org/projects/kurnia-ilham>
- [3] S. Fatimah *et al.*, "Development of an Ultrasonic Pest Repellent System Based on ESP32 and the Internet of Things (IoT)," *Justek : Jurnal Sains*

- dan Teknologi*, vol. 9, no. 1, p. 85, Mar. 2026, doi: 10.31764/justek.v9i1.37949.
- [4] M. A. Ali, R. K. Dhanaraj, and S. Kadry, "AI-enabled IoT-based pest prevention and controlling system using sound analytics in large agricultural field," *Comput. Electron. Agric.*, vol. 220, p. 108844, May 2024, doi: 10.1016/j.compag.2024.108844.
- [5] J. C. De Dios Garcia *et al.*, "Comparative study of Convolutional Neural Networks performance and efficiency with YOLOv8 models applied for pest detection purposes in bean plants.," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 16, no. 2, pp. 112–122, Mar. 2025, doi: 10.61467/2007.1558.2025.v16i2.603.
- [6] A. Raza, M. K. Shaikh, O. Ahmed Siddiqui, A. Ali, and A. Khan, "Enhancing Agricultural Pest Management with YOLO V5: A Detection and Classification Approach," *UMT Artificial Intelligence Review*, vol. 3, no. 2, Dec. 2023, doi: 10.32350/umt-air.32.02.
- [7] Y. Wang, M. Wu, and Y. Shen, "Identifying the Growth Status of Hydroponic Lettuce Based on YOLO-EfficientNet," *Plants*, vol. 13, no. 3, p. 372, Jan. 2024, doi: 10.3390/plants13030372.
- [8] F. Marleny, M. Fitriansyah, Sa'adah, W. A. N. Saputri, R. Ansari, and Mambang, "Segmentasi Citra Keretakan Dinding Beton Menggunakan Teknik Perbandingan Evaluasi Metrik," *TEMATIK*, vol. 10, no. 1, pp. 28–33, Jun. 2023, doi: 10.38204/tematik.v10i1.1261.
- [9] N. Nagaraj, R. Likhitha, B. N. Indu, and R. V. Manjunath, "Automated approach for wildlife detection and tracking using YOLOv8 deep learning algorithm," *Discover Electronics*, vol. 2, no. 1, p. 107, Dec. 2025, doi: 10.1007/s44291-025-00126-2.
- [10] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, "Label Studio: Data labeling software," GitHub, 2020. [Online]. Available: <https://github.com/HumanSignal/label-studio>
- [11] G. Jocher, Q. Jing, and A. Chaurasia, "Ultralytics YOLO," GitHub, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [12] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [13] A. L. D. Araújo *et al.*, "Two-step pipeline for oral diseases detection and classification: a deep learning approach," *Frontiers in Oral Health*, vol. 6, Oct. 2025, doi: 10.3389/froh.2025.1659323.
- [14] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019. [Online]. Available: <https://arxiv.org/abs/1905.11946>
- [15] Y. Zhu, Y. Zhao, Y. He, B. Wu, and X. Su, "YOLO-WildASM: An Object Detection Algorithm for Protected Wildlife," *Animals*, vol. 15, no. 18, p. 2699, Sep. 2025, doi: 10.3390/ani15182699.