

Exploration of RTMDet and SAHI for Long-Tailed Macaque Damage in Corn Fields

Yunnerja

Department of Informatics
Universitas Muhammadiyah Banjarmasin
Banjarmasin, Kalimantan Selatan
yunnerja_2255201110021@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—Attacks by long-tailed macaques (*Macaca fascicularis*) on corn farmland cause a drastic decline in yield from 7–9 metric tons to only 4–5 metric tons per hectare. Traditional control methods have proven ineffective, necessitating solutions based on smart technology. This study explores the RTMDet (Real-Time Detection Transformer) algorithm combined with the SAHI (Slicing Aided Hyper Inference) framework to evaluate wildlife detection performance. Four RTMDet variants RTMDet-tiny, RTMDet-s, RTMDet-m, and RTMDet-l were tested using a dataset of 2,271 COCO annotated images. Evaluation on the test set showed that the standalone RTMDet-s achieved the best performance with a mAP@0.5 of 0.9070, a Recall of 0.9585, and an F1-Score of 0.9321. Applying SAHI reduced accuracy and caused computational delays (down to 3.02 FPS) because the monkey objects occupy a relatively large area within the image frame. Based on these results, the standard RTMDet-s is highly recommended for integration as a real-time automated trigger system within IoT-based wildlife deterrent devices to instantly mitigate attacks and protect farmers' crop yields.

Keywords—RTMDet, SAHI, object detection, long-tailed macaque, Internet of Things.

I. INTRODUCTION

Corn is a strategic agricultural commodity that plays a vital role in Indonesia's national food security, serving as a food source, animal feed, and industrial raw material. However, corn productivity is threatened by wildlife disturbance, which reduces crop yields and directly affects farmers' economic stability [1]. Technology-based automatic detection systems have become an increasingly common solution to this problem [2].

Based on observations and interviews with Tunas Tani partners, the primary threat to corn cultivation in the area is attacks by long-tailed macaques, which are especially damaging because they occur in large groups throughout the entire growth cycle, from planting to harvest [3]. As a result, the partners have lost 3 to 4 metric tons of corn per hectare [4]. Current control measures manual deterrence with slingshots, stones, gunshots, and traps, as well as nets, flares, and loud noises have proven ineffective, since they require constant human presence, consume time and labor, and only temporarily and partially keep monkeys away [5]. This underscores the need for a more effective, efficient, and sustainable solution.

Image-based object detection has been widely applied in agriculture to identify plant diseases and wildlife disturbances [6], [7], [8]. A prior IoT-based deterrent

prototype used YOLOv3 Tiny to detect long-tailed macaques, but its limited accuracy and tendency to confuse monkeys with other objects, such as humans, risked triggering false positives that could disrupt human activity near farmland. This study addresses that limitation by exploring RTMDet, a newer, next-generation detector that outperforms YOLOv3 Tiny and achieves 52.8% AP on COCO at over 300 FPS across four size variants (tiny, s, m, l) [9], offering a better balance of speed and accuracy for real-time deployment. RTMDet is further combined with SAHI, a slicing-based inference framework shown to improve detection accuracy by up to 6.8% AP on small objects in aerial imagery [10]. However, SAHI's effectiveness on large, close-range objects such as macaques in real field conditions as opposed to the small, distant objects it was originally designed for has not previously been evaluated, and this gap forms the second contribution of this study.

Accordingly, this study makes two main contributions: a systematic comparison of four RTMDet variants for detecting long-tailed macaques, evaluating their accuracy, speed, and suitability for real-time IoT deployment, going beyond the earlier YOLOv3 Tiny approach, and the first evaluation of SAHI's effectiveness on large, near-range wildlife objects, in contrast to its established use on small objects, together with an identification of the optimal SAHI parameter configuration for this dataset. Together, these contributions are expected to yield a model configuration that can be integrated into monkey-repellent hardware, resulting in a smarter, more effective, and sustainable agricultural protection system.

II. METHODS

This study is a computational experimental study aimed at exploring the performance of the RTMDet algorithm across various model sizes (tiny, small, medium, large) for detecting long-tailed macaques in digital images. The approach used is a quantitative one, comparing standard COCO evaluation metrics including mean Average Precision (mAP), Precision, Recall, and F1-Score across each tested configuration [11]. In addition to comparing the different model variants, this study also examined the impact of applying the SAHI (Slicing Aided Hyper Inference) framework on accuracy and inference speed across four different parameter configuration scenarios.

The hardware and software used in this study are listed in Table 1 and Table 2. All training and experimentation processes were conducted on the Google Colaboratory

platform, which provides online access to NVIDIA Tesla T4 GPU.

Table 1. Hardware

Component	Specifications
Processor	Google Colab GPU (NVIDIA T4)
RAM	12 GB (Google Colab)
GPU	NVIDIA Tesla T4 (Google Colab)
Platform	Google Colaboratory (Colab)

Table 2. Software Specification

Software	Version / Description
Python	3.10
MMDetection	3.0.0
MMEEngine	0.10.7
MMCV	2.0.1
SAHI	0.12.0
PyTorch	2.0.0+cu118
Roboflow	Dataset annotation platform
Pycocotools	COCO Standard AP Evaluation

The image data used in this study were obtained through direct field photography on Kembang Island, South Kalimantan. Images were captured under various lighting conditions and against diverse backgrounds to increase the dataset's variation and representativeness of real world field conditions. The acquired images cover a wide range of monkey positions, postures, and distances from the camera, ensuring that the trained model possesses strong generalization capabilities.



Figure 1. Image of a Long-Tailed Macaque

All collected images were then uploaded to the Roboflow platform for annotation. Annotation was performed using the bounding box method for each long-tailed macaque object identified in the images. The annotation format used was the COCO (Common Objects in Context) format, which is a common standard in deep learning-based object detection. The annotated dataset was then divided into three subsets: a training set of 1,592 images (70%), a validation set of 453 images (20%), and a test set of 226 images (10%), for a total of 2,271 images. The annotated dataset contains 4,211 bounding-box annotations across 2,271 images, averaging 1.9 macaque objects per image. Images have a median resolution of

3248×1440 pixels (a wide aspect ratio) and an average size of 4.68 megapixels, ranging from 4.68 to 9.37 megapixels. Most images were taken at close-to-medium distance from the macaques, so the annotated monkey objects generally occupy a relatively large proportion of the image area rather than appearing as small, distant objects — a dataset characteristic that later relates directly to the discussion of why the SAHI framework, which is designed primarily for small-object detection, does not improve performance on this dataset.



Figure 2. Annotations of Long-Tailed Macaque Images

This study was conducted in the following stages:

- **Dataset Collection and Preparation.** Long-tailed macaque images were collected by taking photos directly on Kembang Island. The images were then annotated using the Roboflow platform in COCO format and divided into training, validation, and test subsets.
- **Environment Configuration and Installation.** The computing environment was set up on Google Colab by installing the MMDetection framework version 3.0.0 along with its dependencies: MMEEngine, MMCV, PyTorch, SAHI, and pycocotools [12].
- **RTMDet Model Training.** The four RTMDet model variants (tiny, small, medium, large) were trained using the prepared training dataset. The training process was monitored via training loss graphs and mAP values on the validation set at each checkpoint.
- **Evaluation on the Test Set.** The best model from each variant was evaluated on the test set using pycocotools to calculate the following metrics: mAP@0.5, mAP@0.5:0.95, mAP@0.75, Precision, Recall, and F1-Score.
- **Comparison of Inference Without SAHI and With SAHI.** Each model variant was tested using two inference approaches: standard inference without SAHI and inference using the SAHI framework. The results of both approaches were compared based on all evaluation metrics as well as inference speed (FPS).
- **SAHI Parameter Exploration.** Variations were applied to the two main SAHI parameters slice size and overlap ratio to identify the configuration that yields optimal results on the dataset used.

A. Training Configurations and SAHI Parameters

To ensure the validity and consistency of the test results for the object detection algorithm in the presence of long-tailed monkey interference, all experiments were run using standardized computational parameters. The training configurations, including adjustments to the number of epochs, batch size, and learning rate for each RTMDet variant, are shown in Table 3.

Table 3. RTMDet Variant Configurations

Parameter	RTM Det-t	RTM Det-s	RTM Det-m	RTM Det-l
Epoch	50	50	30	40
Batch Size	8	8	8	8
Learning Rate	0.004	0.004	0.004	0.004

Next, testing of SAHI framework parameter combinations including variations in image slice dimensions and overlap ratios is presented in Table 4 as a reference for analyzing the system's inference computation.

Table 4. SAHI Parameter Configurations

SAHI Parameters	Trial 1	Trial 2	Trial 3	Trial 4
Slice Height	512	512	320	256
Slice Width	512	512	320	256
H/W Overlap	0.2	0.4	0.3	0.2
Threshold	0.3	0.3	0.3	0.3

B. Data Analysis

Data analysis was performed quantitatively based on the standard COCO evaluation metrics. The main metrics used are as follows.

- Mean Average Precision (mAP@0.5) measures detection accuracy with an Intersection over Union (IoU) threshold of 0.5. This value is the primary metric used to compare performance across model variants [13].
- mAP@0.5:0.95 is the average of mAP across IoU thresholds ranging from 0.50 to 0.95 in 0.05 increments. This metric is the official COCO evaluation standard and is more stringent than mAP@0.5.
- Precision measures the proportion of true positives among all predictions generated by the model, while Recall measures the proportion of objects successfully detected among all objects actually present in the test data [14]. Comparisons of these evaluation metrics have been widely used in digital image processing research to measure the performance of detection models comprehensively and objectively [15].
- The F1-Score is the harmonic mean of Precision and Recall, used as a measure of the balance between the two, calculated using the following equation:

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (1)$$

- Frames Per Second (FPS) is used to measure the inference speed of each model, particularly when comparing inference without SAHI to inference with SAHI.

All metrics were calculated using the pycocotools library, which is the official implementation of the COCO evaluation standard. Results were compared descriptively through tables and graphs to identify the model variants and SAHI configurations that provided optimal performance on the long-tailed macaque detection dataset used.

III. RESULTS AND DISCUSSION

A. RTMDet Model

Training was performed on four variants of the RTMDet model RTMDet-tiny (t), RTMDet-s, RTMDet-m, and RTMDet-l using the MMDetection framework. Each variant was trained for a tailored number of epochs: RTMDet-t and RTMDet-s were trained for 50 epochs, RTMDet-m for 30 epochs, and RTMDet-l for 40 epochs. RTMDet is designed to achieve an optimal balance between speed and accuracy in real-time inference, with various model size variants available that can be adjusted to suit computational needs.

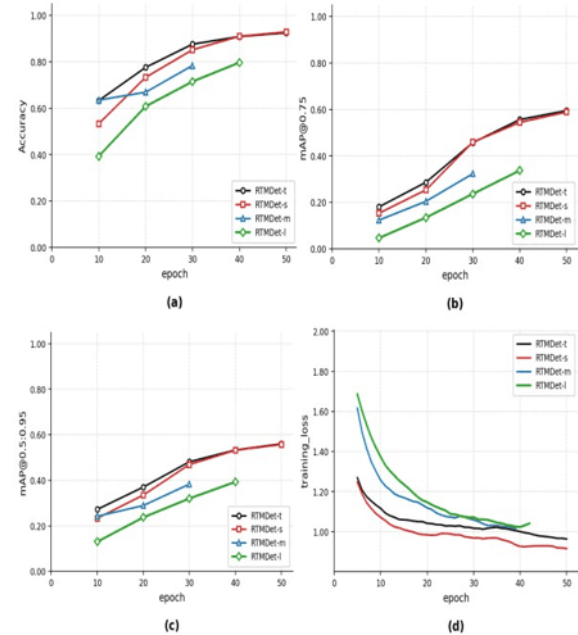


Figure 3. Training Graph of RTMDet Variants

Table 5. RTMDet Variant Results (Validation Set)

Variant	mAP@0.5	mAP@0.75	mAP @0.5:0.95	Training loss
RTMDet-t	0.925	0.595	0.56	0.9538
RTMDet-s	0.929	0.589	0.557	0.9011

RTMDet-m	0.784	0.323	0.383	1.0062
RTMDet-l	0.797	0.337	0.393	1.013

Based on Figure 3 and Table 5, RTMDet-t achieved the highest mAP@0.5 on the validation set at 0.925, followed by RTMDet-s at 0.929, RTMDet-m at 0.784, and RTMDet-l at 0.797. This pattern indicates that smaller models actually produce higher accuracy on this dataset. This can be explained by the relatively limited size of the dataset, meaning that models with fewer parameters are better able to generalize effectively and minimize the risk of overfitting compared to larger model variants. In practical implementation, the strong generalization ability of these smaller models is crucial to ensure that the system does not misidentify other objects in the vicinity of the cornfields (such as leaves or other animals) as monkey disturbances.

The training loss values across all variants show a consistent downward trend as the number of epochs increases, indicating that the training process is proceeding well and that the model is successfully and stably learning the visual features of long-tailed macaques. RTMDet-s recorded the lowest training loss value of 0.5697, indicating the most optimal convergence among the four variants.

Model performance evaluation was conducted on a test set that was completely separate from the training process. Measurements used standard COCO metrics, including mAP@0.5, mAP@0.5:0.95, mAP@0.75, Precision, Recall, and F1-Score. Evaluation on the test set is the most critical stage because it reflects the model's ability to detect objects in data it has never seen before, making the results more representative of the model's performance under real-world conditions.

Table 6. RTMDet Variant Results (Test Set)

Variant	mAP @0.5	mAP @0.75	mAP @0.5:0.95	Precision	Recall	F1-Score
RTMDet-t	0.9062	0.5886	0.5380	0.9062	0.9539	0.9294
RTMDet-s	0.9070	0.5374	0.5302	0.9070	0.9585	0.9321
RTMDet-m	0.8071	0.3283	0.3874	0.8071	0.9309	0.8646
RTMDet-l	0.7959	0.3451	0.3862	0.7959	0.9194	0.8532

Table 6 shows that RTMDet-s achieved the best overall performance on the test set with a mAP@0.5 of 0.9070, a Recall of 0.9585, and an F1-Score of 0.9321. RTMDet-t ranked second with an mAP@0.5 of 0.9062 and an F1-Score of 0.9294. Meanwhile, RTMDet-m and RTMDet-l achieved mAP@0.5s of 0.8071 and 0.7959, respectively. The high Recall value (0.9585) for the RTMDet-s variant provides logical assurance that the protection system has very high sensitivity for detecting monkey herds that sneak in, thereby minimizing the risk of pests slipping past camera surveillance.

RTMDet-s likely outperforms the other variants because its architecture strikes the best balance between model capacity and the size of the training dataset used: it is large enough to capture the distinctive visual features of long-tailed macaques, yet compact enough to avoid overfitting on a dataset of only 2,271 images. In contrast, RTMDet-m and RTMDet-l, with substantially more parameters, appear prone to overfitting on this limited dataset, while RTMDet-t, despite comparable accuracy to RTMDet-s, converges with a slightly higher training loss, indicating marginally less stable learning.

The mAP values for all variants decrease slightly from the validation set to the test set, which is expected since the test set is entirely unseen data and therefore better reflects each model's true generalization performance.

Similarly, the lower mAP@0.5:0.95 compared to mAP@0.5 across all variants is consistent with the stricter, multi-threshold nature of this metric.

B. Inference Without SAHI and With SAHI

The next step is to compare the inference results without SAHI with those using the SAHI (Slicing Aided Hyper Inference) framework across all RTMDet variants. SAHI works by slicing the input image into several overlapping segments, performing inference on each segment independently, and then combining all prediction results. This approach is designed to improve the detection of small objects in high-resolution images.

Table 7. Performance Comparison Without SAHI and With SAHI (512x512 Configuration, 0.2 Overlap)

Variant	Mode	mAP @0.5	mAP@0.5:0.95	Recall	F1-Score	FPS
RTMDet-t	Without Sahi	0.9062	0.5380	0.9539	0.9294	30.65
RTMDet-t	Sahi 512x512	0.8607	0.4966	0.9055	0.8825	23.42
RTMDet-s	Without Sahi	0.9070	0.5302	0.9585	0.9321	29.91
RTMDet-s	Sahi 512x512	0.8843	0.5070	0.9309	0.9070	19.99
RTMDet-m	Unverified	0.8071	0.3874	0.9309	0.8646	22.84
RTMDet-m	Sahi 512x512	0.7185	0.3130	0.8548	0.7808	16.87
RTMDet-l	Unverified	0.7959	0.3862	0.9194	0.8532	14.25
RTMDet-l	Sahi 512x512	0.7290	0.3204	0.8410	0.7810	12.19

Based on Table 7, the application of SAHI with a 512×512 slice configuration and 0.2 overlap on all RTMDet variants resulted in a decrease in the value of the mAP@0.5 compared to inference without SAHI. The largest decrease occurred in RTMDet-l by 0.0669 points,

while the smallest decrease occurred in RTMDet-s by 0.0227 points. This contradicts the common assumption that SAHI always improves detection performance.

This phenomenon can be explained by the characteristics of the dataset used. The images in the long-tailed macaque dataset in this study generally depict subjects that are relatively large compared to the entire image frame. Akyon states that SAHI is most effective when applied to objects with small pixel sizes in an image, such as in aerial or remote sensing imagery. When objects are large, the slicing mechanism actually risks cutting the monkey's body in half, causing important visual features such as body shape and posture to be fragmented across multiple slices, which leads to a decrease in detection accuracy. For managing disturbances in cornfields, this finding is crucial because it demonstrates that using standard inference (without SAHI) is far safer for recognizing the monkey's complete posture from close or medium camera monitoring distances.

In terms of computational speed, the application of the SAHI framework consistently reduces the FPS across all model variants. For example, in the RTMDet-t variant, the inference speed dropped from 30.65 FPS (without SAHI) to 23.42 FPS after applying 512×512 slicing. This decrease aligns with the theory that the process of dividing an image into multiple sub-patches and recombining them using NMS requires processing time.

Further exploration was conducted by varying the two main SAHI parameters slice size and overlap ratio to identify the configuration that yields optimal results on the long-tailed macaque dataset. Four configurations were tested: 512×512 with 0.2 overlap; 512×512 with 0.4 overlap; 320×320 with 0.3 overlap; and 256×256 with 0.2 overlap.

Table 8. Results of SAHI Parameter Exploration (Best Variant: RTMDet-s)

SAHI Configuration	Best Variant	mAP @0.5	mAP @0.5: 0.95	Recall	F1-Score	FPS
512×512, ov 0.2	RTMDet-s	0.8843	0.5070	0.9309	0.9070	19.99
512×512, ov 0.4	RTMDet-s	0.8843	0.5070	0.9309	0.9070	19.16
320×320, ov 0.3	RTMDet-s	0.8154	0.4311	0.8894	0.8508	5.29
256×256, ov 0.2	RTMDet-s	0.7673	0.3688	0.8802	0.8199	3.02



Figure 4. Best Results for RTMDet-s

Based on Table 8, the 512×512 configuration with 0.2 overlap and the 512×512 configuration with 0.4 overlap produced identical values of mAP@0.5 of 0.8843 for RTMDet-s. This similarity in values indicates that increasing the overlap from 0.2 to 0.4 does not provide a significant improvement in accuracy on this dataset, but slightly reduces the FPS from 19.99 to 19.16.

Conversely, using smaller slice sizes (320×320 and 256×256) consistently and significantly reduced detection performance. The 256×256 configuration with 0.2 overlap yields the lowest “mAP@0.5” of 0.7673 and the slowest FPS of 3.02. This phenomenon reinforces the hypothesis that the smaller the slice size, the greater the likelihood that large monkey objects will be cut into fragments too small for the model to accurately recognize them. The drop in FPS to as low as 3.02 FPS at this smallest slice window (256×256) highlights a critical failure if forced onto an automatic deterrent system, as the slow processing delay would allow the monkey to have already damaged the corn crops before the system could send a deterrent signal.

Based on all the exploration results, the 512×512 SAHI configuration with a 0.2 overlap is the best choice when using SAHI, as it results in the smallest drop in accuracy compared to other configurations.

C. General Discussion

Overall, the results of this study show that the RTMDet algorithm is capable of detecting long-tailed macaques (*Macaca fascicularis*) with a high level of accuracy, making it highly reliable as a wildlife disturbance monitoring system in agricultural areas. RTMDet-s proved to be the most optimal variant with a mAP@0.5 of 0.9070, a Recall of 0.9585, and an F1-Score of 0.9321 on the test set. The superiority of RTMDet-s over the larger variants (m and l) is consistent with the finding that, on limited-sized datasets, models with more appropriate capacity tend to produce better generalization.

In the context of corn field protection, the high Recall value (0.9585) of RTMDet-s is the most crucial metric for the system in the field. The behavior of long-tailed macaques, which move in random patterns, requires a highly sensitive detection system to minimize the risk of false negatives (pests escaping camera detection). If the system fails to detect a single lead monkey, other members of the troop will follow, resulting in massive damage to the corn crops. Therefore, RTMDet-s's ability to minimize these detection failures is the cornerstone for reducing the risk of declining crop yields, which have sharply dropped from 7–9 metric tons to just 4–5 metric tons per hectare.

The main finding of this study is that applying the SAHI framework does not improve detection accuracy on the long-tailed macaque dataset used. This is an important contribution because it provides empirical evidence that the effectiveness of SAHI is highly dependent on the size characteristics of objects in the image. As stated by Akyon, the SAHI framework is specifically designed for scenarios where target objects are small in high-resolution images. In the case of monkey disturbances in cornfields, the objects photographed in the field are generally already large within the camera frame (close to medium range). Therefore, standard inference without SAHI remains the superior approach because it is able to capture the visual features of the monkey's body in their entirety without being cropped by the slicing process.

In terms of implementation and system recommendations in the field, RTMDet-s without SAHI achieves the highest inference speed of 29.91 FPS. This speed is more than sufficient for application on IoT hardware for automatic monkey deterrents that require instant, real-time responses. Long-tailed macaque attacks are highly dynamic, fast, and sudden when damaging corn cobs. If the system experiences processing delays (such as when applying SAHI with a small 256 $\times$ 256 window, which drops performance to 3.02 FPS), this delay will allow the macaque to damage the crops before the repeller can be activated. With computational performance reaching nearly 30 FPS and accuracy above 90%, the RTMDet-s model without SAHI is officially recommended for integration into IoT system architectures (such as audiosonic wave transmitters or mechanical repellents) as an efficient automated decision-making "brain" to protect farmland sustainably.

IV. CONCLUSION

This study identifies RTMDet-s without the SAHI framework as the most optimal configuration for detecting long-tailed macaque (*Macaca fascicularis*) disturbances in cornfields, achieving a mAP@0.5 of 0.9070 and an inference speed of 29.91 FPS. This combination of high accuracy and real-time performance makes it well suited as an automatic trigger for IoT-based pest-repelling devices, offering a more effective alternative to traditional control methods. By enabling instant detection and repulsion of monkey herds before damage occurs, this model has the potential to help restore corn yields from the 4–5 metric tons currently seen under macaque disturbance back toward the 7–9 metric tons achievable under normal conditions.

ACKNOWLEDGMENTS

The author would like to thank the academic advisor from the Informatics Study Program at Muhammadiyah University of Banjarmasin for his guidance and direction. Thanks also go to our partner Tunas Tani and the management of the Pulau Kembang Tourist Area for the information, permission, and assistance provided during data collection. Finally, thanks to the entire academic community of Muhammadiyah University of Banjarmasin for their support in providing facilities and a conducive research environment.

REFERENCES

- [1] A. Andriansyah, T. Supartono, and N. Nurdin, "GANGGUAN SATWA LIAR TAMAN NASIONAL GUNUNG CIREMAI TERHADAP LAHAN PERTANIAN DI DESA KARANGSARI KECAMATAN DARMA KABUPATEN KUNINGAN," Nov. 2020. [Online]. Available: <https://journal.uniku.ac.id/index.php/prosiding-fahutan/article/view/3655/2230>
- [2] N. Mamat, M. F. Othman, and F. Yakub, "Animal Intrusion Detection in Farming Area using YOLOv5 Approach," in *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, IEEE, Nov. 2022, pp. 1–5. doi: 10.23919/ICCAS55662.2022.10003780.
- [3] A. A. Mashuri, "Gangguan Monyet Ekor Panjang (*Macaca fascicularis*) pada Kebun Campuran di Kabupaten Kuningan," *Scientific Exploration: Journal of Indonesian Academic Research*, vol. 2, no. 1, pp. 38–51, Nov. 2024, doi: 10.25134/scientificexploration.v2i1.30.
- [4] Mongabay Indonesia, "5 Fakta Monyet Ekor Panjang di Gunungkidul yang Terancam," 2024, [Online]. Available: <https://mongabay.co.id/2024/10/04/5-fakta-monyet-ekor-panjang-di-gunungkidul/>
- [5] I. Darmiyati *et al.*, "Implementasi Sistem Pengusir Hama Monyet Ramah Lingkungan di Kelompok Tani Tunas Harapan Balikpapan," *Jurnal Pengabdian kepada Masyarakat Nusantara*, vol. 6, pp. 5558–5565, 2025.
- [6] C. Lyu *et al.*, "RTMDet: An Empirical Study of Designing Real-Time Object Detectors," Dec. 2022, [Online]. Available: <http://arxiv.org/abs/2212.07784>
- [7] G. Ramya, J. Sreeja, K. Jyothi, J. Radhika, and R. Vigneshwari, "Animal Detection for Crop Protection Using Deep Learning: Insights from YOLO V3, R-CNN, Random Forest," *INTI Journal*, vol. JODS, no. 2025, Jun. 2025, doi: 10.61453/jods.v2025no08.
- [8] R. G. Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.
- [9] F. C. Akyon, S. Onur Altinuc, and A. Temizel, "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection," *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 966–970, Oct. 2022, doi: 10.1109/icip46576.2022.989799.
- [10] P. Singh and R. Krishnamurthi, "IoT-based real-time object detection system for crop protection and agriculture field security," *J. Real. Time.*

- Image Process.*, vol. 21, no. 4, p. 106, Aug. 2024, doi: 10.1007/s11554-024-01488-8.
- [11] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.
- [12] U. Ramadhan *et al.*, “Deteksi Sampah Botol Plastik di Perairan Menggunakan YOLO v4-Tiny,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 7, no. 1, 2025, doi: 10.47233/jteksis.v5i1.1744.
- [13] M. Iqbal, D. M. Midyanti, and S. Bahri, “Deteksi Objek Manusia Pada Citra Menggunakan Single Shot Detector (SSD) Berbasis Edge Computing,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 3, pp. 547–556, Jul. 2024, doi: 10.25126/jtiik.938446.
- [14] M. D. Imansyah and A. Ramadhanu, “Implementasi Algoritma K-Means untuk Klasifikasi Citra Biota Laut: Gurita, Lobster, dan Kerang Laut,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 7, no. 4, pp. 507–513, Oct. 2025, doi: 10.47233/jteksis.v7i4.2271.
- [15] 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.