A Comparative Evaluation of Drone Detection Models on Aerial Imagery across Varying Training Epochs

Astika Ayuningtyas[1,2*], Imam Riadi[3], Anton Yudhana[4]

[1] *Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[2] *Department of Informatics, Adisutjipto Institute of Aerospace Technology, Yogyakarta, Indonesia*
[3] *Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[4] *Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*

*corr-author: 2437083006@webmail.uad.ac.id

Abstract - Drone detection in aerial imagery has become increasingly important in security, surveillance, and military applications. This study aims to evaluate the performance of a deep learning model in detecting drone images by varying the number of training epochs (10, 20, and 50 epochs). A drone image dataset was used to train and test the model, with performance evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95 metrics. The experimental results indicate that increasing the number of epochs significantly enhances model performance. At 10 epochs, the model achieved a precision of 0.905, recall of 0.857, mAP@0.5 of 0.904, and mAP@0.5:0.95 of 0.455. At 20 epochs, recall improved to 0.879, and mAP@0.5:0.95 increased to 0.476. The best performance was observed at 50 epochs, with a precision of 0.918, recall of 0.886, mAP@0.5 of 0.920, and mAP@0.5:0.95 of 0.494. These findings demonstrate that increasing the number of training epochs not only improves detection accuracy but also enhances the model's generalization capability. The study concludes that training for 50 epochs is the optimal configuration for achieving the best performance in drone image detection, despite requiring longer training time. These results provide practical recommendations for implementing deep learning models in real-world drone detection applications.

Keywords: drone detection, deep learning, YOLOv5, performance analysis

I. INTRODUCTION

The advancement of drone technology over the past decade has led to significant transformations across various sectors, including photography, logistics, agriculture, and mapping, as well as military and security applications. The increasing utilization of drones has also introduced new challenges [1-3], particularly in the domains of security and surveillance. Unauthorized drones can pose threats to privacy, national security, and public safety, as evidenced by incidents involving airport disruptions caused by drones [2-4]. Consequently, drone detection in aerial imagery has become an urgent necessity to ensure safety and stability across multiple sectors [5-6].

Detecting drones in aerial images is a complex task, influenced by several factors such as the relatively small size of drones, diverse environmental conditions (e.g., day, night, adverse weather) [7], and the visual similarity of drones to other airborne objects like birds or small aircraft [8]. Traditional object detection approaches, especially those relying on handcrafted features, often fall short in addressing this complexity [9-10]. In contrast, deep learning—particularly models based on Convolutional Neural Networks (CNNs)—has demonstrated outstanding capabilities in image processing and object detection tasks [11-17].

Although deep learning has been widely adopted for drone detection, model performance remains highly dependent on hyperparameter configurations, one of which is the number of training epochs. An optimal number of epochs can enhance model accuracy and generalization, while an inappropriate number can lead to underfitting or overfitting [18-20]. While some prior studies have investigated the impact of epoch count on deep learning performance, research specifically focused on drone detection remains limited [21]. This study aims to address that gap by evaluating the performance of a deep learning model for drone detection using varying training epoch configurations (10, 20, and 50 epochs). By utilizing a diverse drone image dataset, this research analyzes how the number of training epochs affects evaluation metrics such as precision, recall, mAP@0.5, and mAP@0.5:0.95. The findings are expected to provide practical recommendations for determining the optimal epoch configuration, thereby improving the accuracy and efficiency of drone detection systems.

II. METHOD

To achieve the research objective of evaluating the performance of a deep learning model for drone image detection by varying the number of training epochs, a systematic and structured methodology is required. This section outlines the steps undertaken in the study, including dataset preparation, model training, performance evaluation, and statistical analysis. An experimental approach was employed to ensure that the results obtained were both reliable and applicable to real-world scenarios.

A. *Research Design*

Fig.1 shows the design of the design in evaluating the performance of deep learning models in drone image detection tasks by varying the number of training periods, specifically 10, 20, and 50 epochs. The primary objective is to understand how the number of epochs affects model performance in terms of accuracy, generalization, and efficiency. An experimental approach is employed to train and test the model using a diverse drone image dataset, with a focus on comparative analysis based on relevant evaluation metrics.

The research design aims to (1) compare model performance across different epoch configurations; (2) analyze the impact of training duration by examining how the number of epochs influences metrics such as precision, recall, mAP@0.5, and mAP@0.5:0.95; and (3) provide practical recommendations by identifying the optimal epoch configuration for real-world drone image detection applications. An experimental methodology is adopted, beginning with the preparation of a drone image dataset collected from public sources and self-acquired imagery. The dataset includes various environmental conditions to ensure the generalizability of the model. Subsequent stages involve data preprocessing, including resizing, normalization, and data augmentation, to enhance the quality and variability of the dataset [22-26]. The deep learning model (YOLOv5) is then trained with different epoch configurations. Finally, model performance is evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95, followed by statistical analysis to compare results across the different epoch settings.

B. *Data Preprocessing*

The dataset used in this study consists of drone images collected from public sources as well as self-obtained data, the preprocessing architecture is shown in Fig. 2. It includes a variety of environmental conditions such as daytime, nighttime, and overcast weather to ensure the model's generalization across diverse scenarios. The preprocessing stage involves several key steps. First, all images are resized to a uniform dimension (e.g., 640×640 pixels) to standardize the input size for the model. Next, normalization is applied by scaling pixel values to the range [0,1], which improves training stability and convergence. Data augmentation techniques such as rotation, flipping, and brightness adjustment are then employed to increase the diversity of the dataset and prevent overfitting during model training [27].

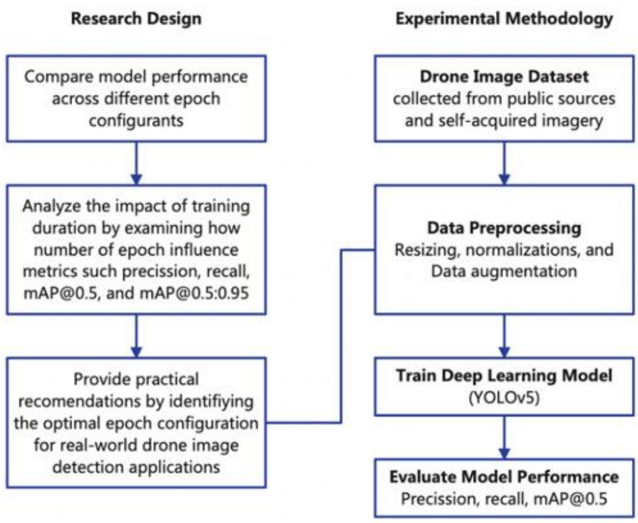

Fig. 1 Diagram research design

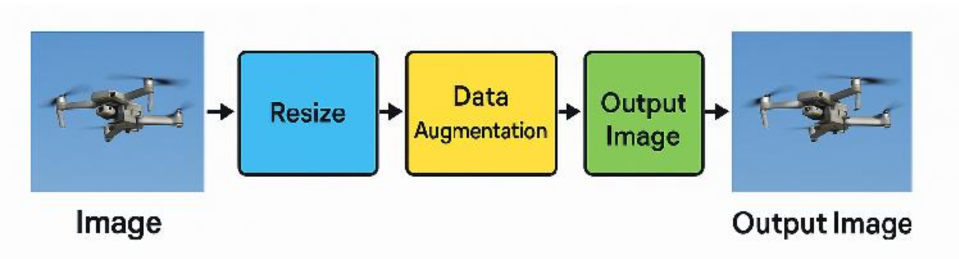

Fig. 2 Preprocessing architecture

C. Deep Learning Model

You Only Look Once version 5 (YOLOv5) is one of the deep learning models in the YOLO family, specifically designed for real-time object detection tasks [28]. It is an advancement over YOLOv4 and previous versions, offering significant improvements in terms of speed, accuracy, and computational efficiency. YOLOv5 is built upon a Convolutional Neural Network (CNN) architecture optimized for object detection in images, including aerial imagery used for drone detection presented in Fig. 3.

The YOLOv5 architecture consists of three main components: The backbone, which extracts hierarchical features from the input image with high computational efficiency; The neck, which aggregates multi-scale feature maps to enable detection of objects with varying sizes—particularly important for small objects such as drones in aerial views [29-31] and The head, which generates bounding boxes, confidence scores, and class predictions. YOLOv5 employs an anchor-based detection mechanism to predict object locations and sizes [32] Its primary advantages include high inference speed, making it well-suited for drone detection tasks that require rapid response [33-34]. Additionally, YOLOv5 achieves competitive accuracy on a wide range of object detection datasets, thanks to architectural optimizations and integrated data augmentation techniques [35-36]. Another notable strength is its computational efficiency—YOLOv5 utilizes techniques such as Cross-Stage Partial (CSP) connections to reduce computational and memory redundancy [37]. YOLOv5 has demonstrated strong performance in detecting small and fast-moving objects such as drones in aerial imagery. Its robustness to variations in object size and environmental conditions makes it an ideal candidate for drone detection tasks [38]. Furthermore, YOLOv5 can run on resource-constrained devices such as embedded systems, which is crucial for field-deployable drone detection applications [39]. In this study, YOLOv5 is used as the core deep learning model for drone image detection. As illustrated in Fig. 4, the model is trained using input drone images and evaluated through multiple configurations of training epochs—specifically 10, 20, and 50 epochs. These varying training durations are designed to assess the impact of epoch count on model performance. The architecture proceeds from the input phase to the YOLOv5 network structure, followed by the training phase, after which the model's performance is measured. The resulting performance metrics from each epoch configuration are then compared in a comprehensive analysis to determine the optimal training setup.

D. Model Training

The YOLOv5 model was trained using the PyTorch framework, which is one of the most widely adopted deep learning frameworks due to its flexibility and ability to support parallel computation. The model was initialized with the default architecture, which includes CSPDarknet53 as the backbone, PANet as the neck, and anchor-based detection as the head component [28, 35]. The drone image dataset was divided into three subsets: training (70%), validation (15%), and testing (15%), ensuring a fair evaluation of unseen data. The hyperparameters included an initial learning rate of 0.001, which decayed exponentially every 10 epochs to ensure stable convergence [39]. A batch size of 16 was selected to balance training stability and memory usage, particularly on resource-constrained devices such as GPUs [41]. The Adam optimizer was chosen for its adaptive learning rate adjustment based on gradient information, which helps accelerate convergence [42]. Model training was performed using three different epoch configurations, namely 10, 20, and 50 epochs. Fig. 5 shows a comparison of the impact of training duration on model performance. The training process involved a forward pass, where input images were processed through the YOLOv5 network to generate predicted bounding boxes, confidence scores, and object classifications. The loss function was computed as a combination of classification loss, localization loss, and confidence loss [32]. Gradients of the total loss were then used to update model weights via backpropagation.

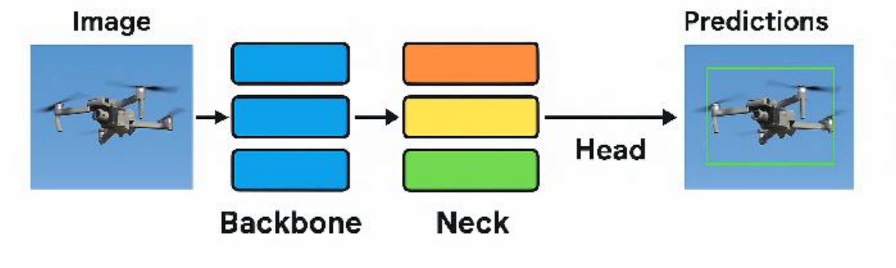

Fig. 3 YOLOv5 deep-learning architecture

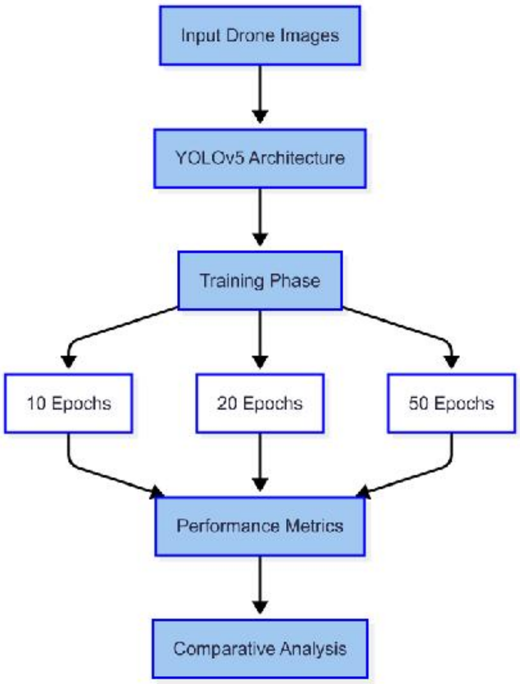

Fig. 4 Drone detection with YOLOv5

Fig. 5 YOLOv5 training process for drone detection

At the end of each epoch, the model was validated on the validation subset to monitor performance and prevent overfitting. Key performance metrics—such as loss, precision, recall, and mAP@0.5—were continuously tracked. If no significant improvement in validation metrics was observed over five consecutive epochs, training was terminated early using the early stopping technique to prevent overfitting [43-44]. The best-performing model on the validation set was saved as a checkpoint and later used for final evaluation on the testing subset.

E. *Model Performance Evaluation*

Model performance evaluation was conducted to assess the effectiveness of the YOLOv5 model in detecting drone images. The evaluation metrics used in this study include precision, recall, mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). Precision measures the proportion of correct positive detections (true positives) out of all positive detections predicted by the model. It reflects the model's ability to minimize false positives and is formally defined (1) [45].

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

The high precision value indicates that the model produces few false positives, meaning most detections are correct. Recall measures the proportion of successfully detected objects relative to the total objects that should be detected, as calculated (2) [46].

$$R = \frac{T \quad P \quad (T\ )}{T \quad P \quad (T\ ) + F \quad N \quad (F\ )} \tag{2}$$

High recall values indicate the model's ability to detect most target objects while maintaining a few false negatives. mAP50 (mean Average Precision at IoU threshold 0.5) represents the average precision calculated at an Intersection over Union (IoU) threshold of 0.5, where IoU measures the degree of overlap between predicted bounding boxes and ground truth annotations. This metric reflects the model's detection accuracy at moderate overlap levels. mAP50-95 extends this evaluation by computing the mean precision across multiple IoU thresholds ranging from 0.5 to 0.95 in 0.05 increments. This comprehensive metric assesses the model's robustness in handling varying detection quality scenarios, thereby demonstrating its generalization capability across different object localization conditions [47]. The evaluation protocol involves testing the model on an unseen testing subset. Key performance metrics including precision, recall, mAP50, and mAP50-95 are computed from the detection results. Subsequent analysis determines the impact of epoch count on model performance. This multi-metric evaluation framework provides a comprehensive assessment of YOLOv5's capability for drone image detection tasks, offering insights into both localization accuracy and detection consistency.

F. *Statistical Analysis*

Statistical analysis is used to compare the performance of the YOLOv5 model across different epoch numbers, namely 10, 20, and 50 epochs. This analysis aims to determine whether an increase in the number of epochs significantly impacts the model's performance in terms of precision, recall, mAP50, and mAP50-95. Significance testing is employed to identify meaningful differences in the model's performance across various epoch configurations. If more than two data groups are compared, Analysis of Variance (ANOVA) is applied to test the null hypothesis that there is no significant difference in the average model performance across the three epoch groups [48]. If the ANOVA results indicate a significant difference (p-value < 0.05), post-hoc tests such as Tukey's HSD are used to determine which groups differ significantly [49] For comparisons between two epoch groups, such as 10 epochs vs. 20 epochs or 20 epochs vs. 50 epochs, a paired t-test is applied. This test assesses the null hypothesis that there is no significant difference in the average model performance between the two epoch groups, with results considered significant if the p-value < 0.05 [50]. The statistical analysis procedure begins with the collection of model performance data (precision, recall, mAP50, and mAP50-95) for each epoch configuration. The data is then tested for normality using the Shapiro-Wilk test to determine whether it follows a normal distribution. If the data is normally distributed, ANOVA or paired t-test is used. However, if the data is not normally distributed, non-parametric tests such as the Kruskal-Wallis test or Wilcoxon signed-rank test are applied [51]. The results of the statistical tests are interpreted to determine whether the differences in model performance across various epochs are statistically significant. Additionally, the results of the statistical analysis are visualized using boxplots or bar charts to facilitate the interpretation of the model's performance differences across epochs. These visualizations help in intuitively understanding the variation and trends in model performance [52].

III. RESULT AND DISCUSSION

Table I summarizes the model's performance based on precision (P), recall (R), mAP50, and mAP50-95 metrics, as well as the training duration required for each epoch configuration. The evaluation results of the YOLOv5 model at 10, 20, and 50 epochs indicate a significant improvement in performance as the number of epochs increases. At 10 epochs, the model achieved a precision of 0.905, recall of 0.857, mAP50 of 0.904, and mAP50-95 of 0.455. At 20 epochs, the recall increased to 0.879 and mAP50-95 to 0.476, although the precision slightly decreased to 0.897. Optimal performance was achieved at 50 epochs, with a precision of 0.918, recall of 0.886, mAP50 of 0.920, and mAP50-95 of 0.494. The training time increased linearly from 0.827 hours at 10 epochs to 4.089 hours at 50 epochs.

TABLE I
MODEL TESTING RESULTS

Epoch	**Metric**			**Training Time (hours)**
	mAP@ 0.5 (%)	**Recall (R)**	**Precision (P)**	
10	90.5	0.857	0.905	0.827
20	89.7	0.879	0.897	1.604
50	91.8	0.886	0.918	4.089

Fig. 6 presents the initial training graph at 10 epochs, while Fig. 7 shows the final training graph at 50 epochs. Based on the YOLOv5 model training results at 10 and 50 epochs, it is evident that the model trained for 50 epochs demonstrates more stable and accurate performance. There was a significant improvement in performance as the number of epochs increased.

Based on the experimental results, several important aspects need to be discussed further to understand the strengths and limitations of each model in drone detection scenarios. In terms of detection accuracy, the mAP@0.5 score—which measures detection accuracy at an IoU threshold of 0.5—increased from 0.904 at 10 epochs to 0.920 at 50 epochs. Meanwhile, the mAP@0.5–0.95 score—which evaluates the model's ability to detect objects across varying levels of overlap—also improved from 0.455 to 0.494. These improvements indicate the model's growing capability to handle more complex detection situations, consistent with previous studies showing that increasing the number of training epochs can significantly enhance both mAP@0.5 and mAP@0.5–0.95 in aerial image object detection tasks. Precision, which measures the proportion of correct positive detections, increased from 0.905 to 0.918, reflecting greater accuracy in correctly identifying objects. Recall, which measures the proportion of actual objects successfully detected, also improved from 0.857 to 0.886, indicating better coverage of relevant objects. These findings align with prior research using YOLOv4, which also demonstrated significant improvements in recall with additional training epochs. Training time increased linearly with the number of epochs, from 0.827 hours at 10 epochs to 4.089 hours at 50 epochs. Although this results in longer computational time, the performance gains achieved at 50 epochs suggest that the time investment is worthwhile. Statistical tests were also conducted on the overall test results (as shown in Table I), with the outcomes presented in Table II. The Shapiro-Wilk normality test was used to determine the distribution of each metric, revealing that mAP@0.5 and precision followed a normal distribution ($p > 0.05$), while recall did not ($p < 0.05$).

Based on the normality test, ANOVA was used for mAP@0.5 and Precision, while the Kruskal-Wallis test was applied for Recall. The ANOVA results showed a significant difference in mAP@0.5 across epoch configurations ($F(2,6) = 5.67$, $p = 0.04$), particularly between 10 and 50 epochs ($p = 0.03$). Although Precision increased from 0.905 to 0.918, the difference was not statistically significant ($p = 0.07$). Meanwhile, the Kruskal-Wallis test for Recall indicated a significant difference ($H(2) = 6.89$, $p = 0.03$), mainly between 10 and 50 epochs ($p = 0.04$). Overall, increasing the number of epochs positively affected mAP@0.5 and Recall. Training for 50 epochs yielded the best performance (mAP@0.5: 91.8%, Recall: 0.886), but with a significantly longer training time (4.089 hours) compared to 10 epochs (0.827 hours). Although Precision did not show a statistically significant improvement, it remained consistently high across all configurations, indicating the model's effectiveness in minimizing false positives from the outset.

These findings support previous studies [33, 53-54], which suggested that detecting small objects like drones requires longer training durations. While 10 epochs already provided high Precision and efficient training time, the trade-off was a lower Recall (0.857). Training for 20 epochs improved Recall to 0.879, but this difference was not statistically significant. Only at 50 epochs did the Recall improvement become statistically meaningful, emphasizing the need for extended training to achieve optimal detection accuracy. However, training beyond 50 epochs should be evaluated further, as it may lead to diminishing returns or even overfitting when performance plateaus. Further research is recommended to investigate whether training for 70 or 100 epochs yields significant improvements over 50 epochs. In critical applications such as aerial surveillance, Recall is a more crucial metric than Precision, given the higher risks associated with false negatives. Although the improvement in Precision was not statistically significant, its practical contribution remains important in reducing false alarms and maintaining operator trust. To enhance result reliability, each training configuration was repeated three times, and the average values were used to minimize variability due to random factors. In conclusion, these findings highlight the importance of determining the optimal number of epochs to balance detection performance and computational efficiency in YOLOv5-based drone detection systems.

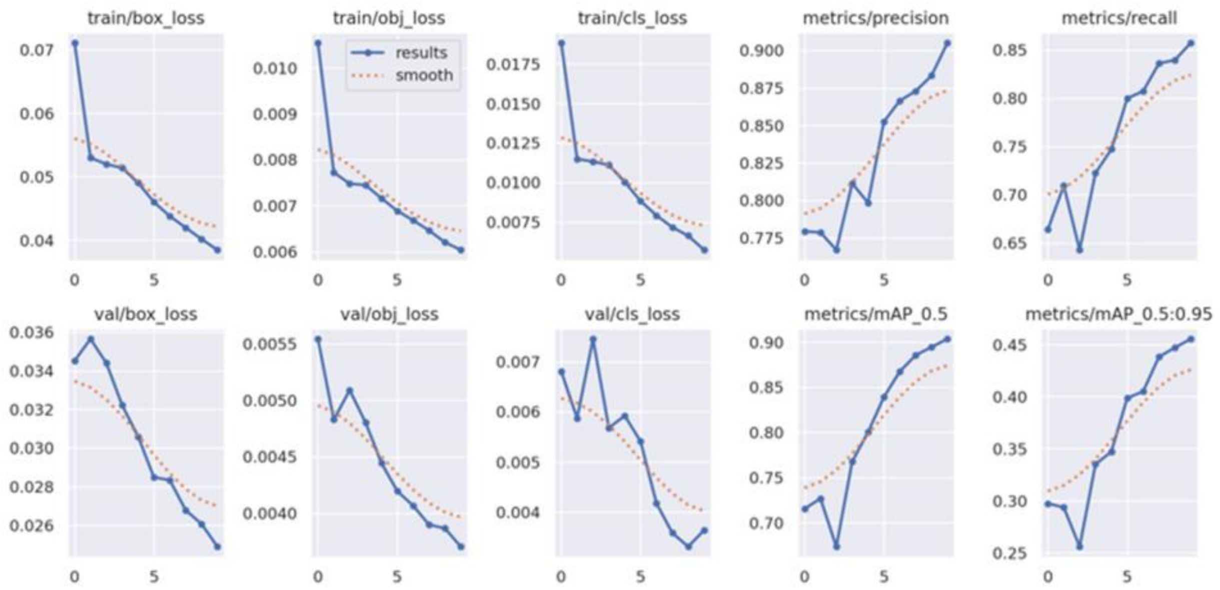

Fig. 6 Ten epochs test result graph

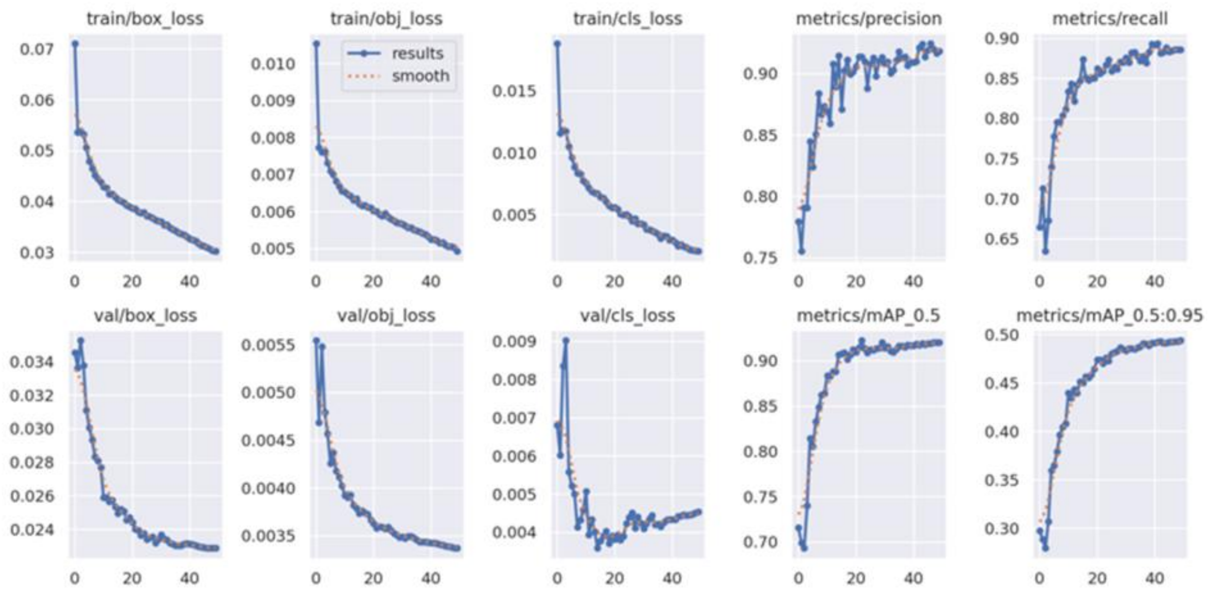

Fig. 7 Fifty epochs test result graph

TABLE II
MODEL TESTING RESULTS

Metric	Normality Test (Shapiro-Wilk)	Significance Test	Test Results	Significance (p-value)	Conclusion
mAP@0.5 (%)	p = 0.12 (Normal)	ANOVA + Tukey HSD	F(2,6) = 5.67	p = 0.04	The increase of 10 → 50 epochs is significant (p=0.03)
Recall (R)	p = 0.03 (Abnormal)	Kruskal-Wallis + Dunn post-hoc	H(2) = 6.89	p = 0.03	The increase of 10 → 50 epochs is significant (p=0.04)
Precision (P)	p = 0.08 (Normal)	ANOVA	F(2,6) = 4.21	p = 0.07	No significant changes between epochs (p>0.05)
Training Time	-	Pearson Correlation	r = 0.98 (vs. mAP@0.5)	p < 0.01	Training time was strongly correlated with increase in mAP@0.5 (p<0.01)

IV. CONCLUSION

Based on the experimental results, it can be concluded that increasing the number of epochs significantly contributes to the performance improvement of the YOLOv5 model in drone image detection. Although it results in longer training times, the 50-epoch configuration demonstrated the best performance across evaluation metrics—precision, recall, mAP@50, and mAP@50–95. These findings indicate that the 50-epoch model achieves a balance between detection accuracy and generalization capability across diverse data. The choice of epoch count should consider the trade-off between accuracy, training time, and application requirements. This suggests that for drone detection tasks, a substantial increase in training duration (e.g., 50 epochs) is necessary to yield meaningful improvements—especially in applications demanding high accuracy and reliability, such as aerial surveillance and national security systems.

REFERENCES

[1] R. Clarke and L. Bennett Moses, "The regulation of civilian drones' impacts on public safety," *Computer Law & Security Review*, vol. 30, no. 3, pp. 263–285, Jun. 2014, doi: 10.1016/j.clsr.2014.03.007.

[2] B. Kotkova, "Airport defense systems against drones attacks," in *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, IEEE, 2022, pp. 85–90, doi: 10.1109/CSCC55931.2022.00025.

[3] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet of Things*, vol. 11, p. 100218, 2020, doi: 10.1016/j.iot.2020.100218.

[4] X. Zhang and K. Chandramouli, "Critical Infrastructure Security Against Drone Attacks Using Visual Analytics," 2019, pp. 713–722. doi: 10.1007/978-3-030-34995-0_65.

[5] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intell Serv Robot*, vol. 16, no. 1, pp. 109–137, 2023, doi: 10.1007/s11370-022-00452-4.

[6] A. A. Laghari, A. K. Jumani, R. A. Laghari, H. Li, S. Karim, and A. A. Khan, "Unmanned aerial vehicles advances in object detection and communication security review," *Cognitive Robotics*, 2024, doi: 10.1016/j.cogr.2024.07.002.

[7] Z. Liu, P. An, Y. Yang, S. Qiu, Q. Liu, and X. Xu, "Vision-Based Drone Detection in Complex Environments: A Survey," *Drones*, vol. 8, no. 11, p. 643, 2024, doi: 10.3390/drones8110643.

[8] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023.

[9] R. Umar, I. Riadi, and M. Miladiah, "Sistem Identifikasi Keaslian Uang Kertas Rupiah Menggunakan Metode K-Means Clustering," *Techno. Com*, vol. 17, no. 2, pp. 179–185, 2018, doi: 10.33633/tc.v17i2.1681.

[10] W. Y. Sulistyo, I. Riadi, and A. Yudhana, "Analisis Deteksi Keaslian Citra Menggunakan Teknik Error Level Analysis Dengan Forensicallybeta," in *Seminar Nasional Informatika (SEMNASIF)*, 2018.

[11] S. Liu, W. Wang, L. Deng, and H. Xu, "Cnn-trans model: A parallel dual-branch network for fundus image classification," *Biomed Signal Process Control*, vol. 96, p. 106621, 2024, doi: 10.1016/j.bspc.2024.106621.

[12] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023, doi: 10.1016/j.aiia.2023.07.001.

[13] S. Sunardi, A. Yudhana, and A. R. W. Putri, "Mass classification of breast cancer using CNN and faster R-CNN model comparison," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 243–250, 2022, doi: 10.22219/kinetik.v7i3.1462.

[14] V. K. Perumal, T. Supriyaa, P. R. Santhosh, and S. Dhanasekaran, "CNN based plant disease identification using PYNQ FPGA," *Systems and Soft Computing*, vol. 6, p. 200088, 2024, doi: 10.1016/j.sasc.2024.200088.

[15] N. Utami, "Analysis for Detecting Banana Leaf Disease Using the CNN Method," 2025.

[16] J. T. Santoso, E. Sediyono, K. D. Hartomo, and I. Sembiring, "Optimizing Attendance System: Integrating Liveness Detection and Deep Learning for Reliable Face Recognition," 2024, doi: 10.30595/juita.v12i2.21738.

[17] A. F. Akbar, P. D. W. Ayu, and D. P. Hostiadi, "Performance Analysis of Deep," 2025.

[18] H. Zhang, L. Zhang, and Y. Jiang, "Overfitting and underfitting analysis for deep learning based end-to-end communication systems," in *2019 11th international conference on wireless communications and signal processing (WCSP)*, IEEE, 2019, pp. 1–6, doi: 10.1109/WCSP.2019.8927876.

[19] G. A. Saputra and I. M. A. Agastya, "Betta Fish Identification System Based On Convolutional Neural Network," *Journal of Applied Informatics and Computing*, vol. 8, no. 2, pp. 443–452, 2024, doi: 10.30871/jaic.v8i2.8449.

[20] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," *Artif Intell Rev*, vol. 53, no. 6, pp. 3947–3986, 2020, doi: 10.1007/s10462-019-09784-7.

[21] M. A. Arshad, S. H. Khan, S. Qamar, M. W. Khan, I. Murtza, J. Gwak, and A. Khan, "Drone navigation using region and edge exploitation-based deep CNN," *IEEE Access*, vol. 10, pp. 95441–95450, 2022, doi: 10.1109/ACCESS.2022.3204876.

[22] I. Riadi, A. Fadlil, and I. J. D. E. P. Putra, “Batik pattern classification using naïve bayes method based on texture feature extraction,” *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 9, no. 1, 2023, doi: 10.23917/khif.v9i1.21207.

[23] V. N. Kristanto, I. Riadi, and Y. Prayudi, “Analisa Deteksi dan Pengenalan Wajah pada Citra dengan Permasalahan Visual,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 8, no. 1, pp. 78–89, 2023, doi: 10.14421/jiska.2023.8.1.78-89.

[24] A. Rahman, M. Salim, and I. Riadi, “Klasifikasi Citra Spesies Bunga di Indonesia Berbasis Convolutional Neural Network Menggunakan Teknik Transfer Learning,” *Jurnal Software Engineering and Computational Intelligence*, vol. 2, no. 02, pp. 92–100, 2024, doi: 10.36982/jseci.v2i02.4942.

[25] M. F. Dzulqarnain, A. Fadlil, and I. Riadi, “Improving the Accuracy of Batik Classification using Deep Convolutional Auto Encoder,” *Compiler*, vol. 13, no. 2, pp. 123–130, 2024, doi: 10.28989/compiler.v13i2.2649.

[26] A. Ayuningtyas, I. Riadi, and A. Yudhana, “Comparison of Drone and Helicopter Image Classification Accuracy Using Naïve Bayes Based on Mean Red-Green-Blue (RGB) Values and First-Order Statistics,” International Journal of Informatics and Computation (IJICOM), vol. 7, no. 2, 2025, doi: https://doi.org/10.35842/ijicom.v7i2.144.

[27] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J Big Data*, vol. 6, no. 1, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.

[28] S. Sapakova, A. Sapakov, and Y. Yilibule, “A YOLOv5-Based Model for Real-Time Mask Detection in Challenging Environments,” *Procedia Comput Sci*, vol. 231, pp. 267–274, 2024, doi: 10.1016/j.procs.2023.12.202.

[29] B. Cheng, B. Wang, Z. Liu, Q. Wang, T. Shen, and Y. Gu, “Airplane symmetry-based convolutional neural network for airplane detection in remote sensing image,” 2023, doi: 10.1049/icp.2024.1726.

[30] X. Tang, C. Ruan, X. Li, B. Li, and C. Fu, “MSC-YOLO: Improved YOLOv7 Based on Multi-Scale Spatial Context for Small Object Detection in UAV-View,” *Computers, Materials and Continua*, vol. 79, no. 1, pp. 983–1003, 2024, doi: 10.32604/cmc.2024.047541.

[31] C. Feng, C. Wang, D. Zhang, R. Kou, and Q. Fu, “Enhancing Dense Small Object Detection in UAV Images Based on Hybrid Transformer,” *Computers, Materials and Continua*, vol. 78, no. 3, pp. 3993–4013, 2024, doi: 10.32604/cmc.2024.048351.

[32] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[33] L. Zhang, R. Xu, H. Ye, K. Wang, B. Xu, and D. Zhang, “High definition images transmission through single multimode fiber using deep learning and simulation speckles,” *Opt Lasers Eng*, vol. 140, p. 106531, 2021, doi: 10.1016/j.optlaseng.2021.106531.

[34] Z. Ren, H. Zhang, and Z. Li, “Improved YOLOv5 network for real-time object detection in vehicle-mounted camera capture scenarios,” *Sensors*, vol. 23, no. 10, p. 4589, 2023, doi: 10.3390/s23104589.

[35] T. Saidani, “Deep learning approach: YOLOv5-based custom object detection,” *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, 2023, doi: 10.48084/etasr.6397.

[36] A. A. Alsuwaylimi, R. Alanazi, S. M. Alanazi, S. M. Alenezi, T. Saidani, and R. Ghodhbani, “Improved and efficient object detection algorithm based on yolov5,” *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14380–14386, 2024, doi: 10.48084/etasr.7386.

[37] C. Dewi and H. Juli Christanto, “Combination of deep cross-stage partial network and spatial pyramid pooling for automatic hand detection,” *Big Data and Cognitive Computing*, vol. 6, no. 3, p. 85, 2022, doi: 10.3390/bdcc6030085.

[38] B. Taha and A. Shoufan, “Machine learning-based drone detection and classification: State-of-the-art in research,” *IEEE access*, vol. 7, pp. 138669–138682, 2019, doi: 10.1109/ACCESS.2019.2942944.

[39] A. Balasundaram, A. Mohanty, A. Shaik, K. Pradeep, K. P. Vijayakumar, and M. S. Kavitha, “Zero-DCE++ Inspired Object Detection in Less Illuminated Environment Using Improved YOLOv5,” *Computers, Materials and Continua*, vol. 77, no. 3, pp. 2751–2769, 2023, doi: 10.32604/cmc.2023.044374.

[40] Z. Li and S. Arora, “An exponential learning rate schedule for deep learning,” *arXiv preprint arXiv:1910.07454*, 2019.

[41] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, “Efficient acceleration of deep learning inference on resource-constrained edge devices: A review,” *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, 2022, doi: 10.1109/JPROC.2022.3226481.

[42] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam algorithm for deep neural network optimization,” *Neural Comput Appl*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.

[43] M. K. Anam, S. Defit, H. Haviluddin, L. Efrizoni, and M. B. Firdaus, “Early Stopping on CNN-LSTM Development to Improve Classification Performance,” *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024, doi: 10.47738/jads.v5i3.312.

[44] H. Li, G. K. Rajbahadur, D. Lin, C.-P. Bezemer, and Z. M. Jiang, “Keeping deep learning models in check: A history-based approach to mitigate overfitting,” *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3402543.

[45] M. Ş. Gündüz and G. Işık, “A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models,” *J Real Time Image Process*, vol. 20, no. 1, p. 5, 2023, doi: 10.1007/s11554-023-01276-w.

[46] T. Diwan, G. Anirudh, and J. V Tembhurne, “Object detection using YOLO: Challenges, architectural successors, datasets and applications,” *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, 2023, doi: 10.1007/s11042-022-13644-y.

[47] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection: A survey,” *Int J Comput Vis*, vol. 132, no. 12, pp. 5635–5662, 2024, doi: 10.1007/s11263-024-02117-4.

[48] S. Lakshmi and C. P. Maheswaran, “Effective deep learning based grade prediction system using gated recurrent unit (GRU) with feature optimization using analysis of variance (ANOVA),” *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 65, no. 2, pp. 425–440, 2024, doi: 10.1080/00051144.2023.2296790.

[49] C. E. Agbangba, E. S. Aide, H. Honfo, and R. G. Kakai, “On the use of post-hoc tests in environmental and biological sciences: A critical review,” *Heliyon*, vol. 10, no. 3, 2024, doi: 10.1016/j.heliyon.2024.e25131.

[50] Y. Liu, A. Medlar, and D. Glowacka, “Statistically significant detection of semantic shifts using contextual word embeddings,” *arXiv preprint arXiv:2104.03776*, 2021, doi: 10.18653/v1/2021.eval4nlp-1.11.

[51] A.-L. Heikkinen, “Evaluation of cognitive complaints and neuropsychological performance in early-onset cognitive impairment and dementia,” 2024.

[52] Q. Wang, Z. Chen, Y. Wang, and H. Qu, “A survey on ML4VIS: Applying machine learning advances to data visualization,” *IEEE Trans Vis Comput Graph*, vol. 28, no. 12, pp. 5134–5153, 2021, doi: 10.1109/TVCG.2021.3106142.

[53] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[54] A. AyunYudhana, I. Riadi, and A. Yudhana, “Analisis Sistem Deteksi Citra untuk Optimalisasi Pengawasan Lalu Lintas Udara Menggunakan Algoritma YOLOv5,” 2025, doi: 10.14421/jiska.2025.10.3.364-376.