International Journal of Applied Mathematics, Sciences, and Technology for National Defense
Volume 2, Issue 3, 111-122
e_ISSN: 2985-9352, p_ISSN: 2986-0776
https://www.journal.foundae.com/index.php/JAS-ND
DOI : https://doi.org/10.58524/app.sci.def..v2i3.439

Unmanned aerial vehicle classification and detection system based on deep learning, internet of military things, and PID control system

M. A. Versantariqh Lesmana*
The Indonesian Navy,
INDONESIA

Iqbal Ahmad Dahlan
Indonesia Defense University,
INDONESIA

Hendrana Tjahjadi
Indonesia Defense University,
INDONESIA

Setya Widyawan Prakosa
National Taiwan University of Science and Technology
TAIWAN

Muhammad Azka Firdaus
The Indonesian Army,
INDONESIA

Article Info

Article history:

Received: July 26, 2024
Revised: October 15, 2024
Accepted: November 18, 2024
Published: December 30, 2024

Keywords:

Air Defense
Flask
MIT APP Inventor
Telegram
YOLOv8

Abstract

Indonesia is an archipelagic country situated between two continents and two oceans. With numerous islands, it is rich in natural resources but faces various military and non-military threats. One significant threat to maritime nations like Indonesia is from the air, which includes direct attacks from manned and unmanned aircraft and using aerial vehicles for intelligence and surveillance. The primary weapon system is crucial for national defense against such threats. Therefore, developing defense equipment in Indonesia must align with technological advancements to ensure quick and efficient operation. This research focuses on creating a classification and reconnaissance system for flying vehicles to enhance air defense capabilities. In the surveillance system, two servos are used for yaw and pitch axes, controlled by a Proportional, Integrative, and Derivative (PID) system. This PID control significantly improves servo movement both dynamically and statically. The system sends notifications via Telegram for monitoring, with an average FPS of 9.6. Flask is used for the website interface, averaging 6.8 FPS, and MIT App Inventor is used for the smartphone interface, averaging 7.6 FPS. This flying vehicle classification and reconnaissance system enhances Indonesia's air defense, utilizing YOLOv8 for classification, PID control for servo movements, and integrated notifications and interfaces for both web and smartphones.

To cite this article: Lesmana, M. A. V., Dahlan, I. A., Tjahjadi, H., Prakosa, S. W., & Firdaus, A. M. (2024). Unmanned aerial vehicle classification and detection system based on deep learning, internet of military things, and PID control system. *International Journal of Applied Mathematics, Sciences, and Technology for National Defense,* 2(3), 115-126.

INTRODUCTION

The Indonesian archipelago comprises over 13,558 islands according to the Central Statistics Agency (BPS) (Andréfouët et al., 2022). It borders Singapore, Malaysia, Papua New Guinea, and Timor-Leste and lies between two continents and two oceans, making it a vital international trade route. This unique geography presents significant challenges for governance, national integration, and security (Khansa et al., 2021). Internally, Indonesia faces issues like the conflict in Papua and diversity and national integration challenges. These internal and external factors present significant challenges in maintaining sovereignty and territorial integrity. One of factors which determining crimes against state sovereignty and territorial integrity include is military aspects, which are interconnected and constantly evolving (Nosach, 2019). While Indonesia's numerous islands make it rich in natural resources, they pose military and non-military threats (Saptono et al., 2022).

***Corresponding Author:**
Azka Versantariqh Lesmana, Indonesia Defense University, Indonesia, Email: azkalesmana8@gmail.com

Military threats are complex, encompassing land, sea, and air dimensions, with air threats being particularly prominent in the current technological era (Roslan, 2023). Operators of military machinery face significant risks during combat operations, and the likelihood of human error increases when targeting enemies on the battlefield (Khan et al., 2021). Air threats to critical infrastructure and security include both manned and unmanned aerial vehicles used for direct attacks, intelligence gathering, and surveillance (Khawaja et al., 2022; Piekarski & Wojtasik, 2022). These vehicles can disrupt communication and navigation systems through electronic warfare, and unmanned aerial vehicles (UAVs) pose potential threats for terrorism and cyberattacks (Ismail et al., 2022). Addressing such threats necessitates enhancing air defense capabilities, developing advanced sensor technologies, and implementing effective strategies to combat electronic warfare and cybersecurity challenges.

Air defence systems, equipped with radar and cameras, are crucial for detecting and mitigating aerial threats. By utilizing computer vision techniques and novel radar processing, outperforming both naive radar-camera combinations and monocular depth estimation (Zhang et al., 2021). Artificial Intelligence (AI) is increasingly important in air defense systems, enabling faster and more precise decision-making (Paliz Ochoa et al., 2022). Advances in automation and artificial intelligence improve real-time object detection and response efficiency (Sunardi et al., 2022). Sensors play a crucial role in detecting, tracking, and identifying airborne threats at long ranges (Huffaker et al., 2020). The Internet of Military Things (IoMT) further enhances situational awareness and connectivity, ensuring rapid and reliable communication across various locations (Bing et al., 2023). Prior research has explored automatic air defence systems using radar, cameras, and neural networks, such as the work by Fazle Rabby Khan et al., which discussed automatic detection and targeting of aerial threats using YOLOv3 and Faster-RCNN, and Ahmad Fahriannur and Meilana Siswanto's research on object tracking using pan-tilt cameras (Fahriannur & Siswanto, 2017; Khan et al., 2021). Earlier research did not employ IoT for real-time monitoring and Telegram notifications. This research expands upon previous work to improve user-friendliness in field applications. It introduces the development of an air defense classification and surveillance system using IoT, YOLOv8, and real-time monitoring through MIT App Inventor to ensure data accuracy and system efficiency.

METHOD

System Design

A system design was created, as shown in Figure 1. Generally, the system has four major components: sensing, understanding, recording, and visualizing and control. Sensing or detecting uses a camera attached to the pitch-yaw servo bracket to detect and scout objects. Understanding involves image processing carried out by a laptop to detect, classify, and scout objects. Additionally, understanding includes a servo motor controller managed by a microcontroller. The recording collects data, which is then stored for further analysis. Finally, visualization displays the system output via a smartphone application. Additionally, there are notifications to facilitate user monitoring of the system (See. Figure 1).

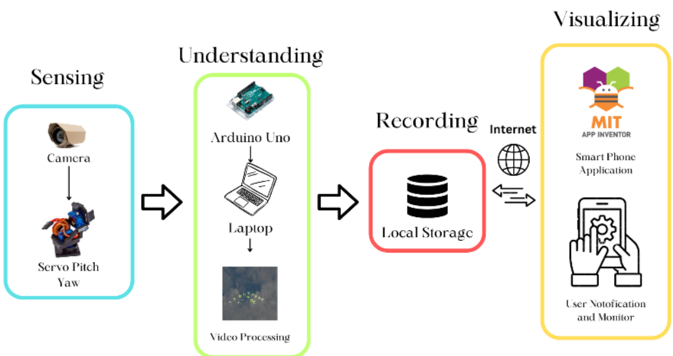

Figure 1. System design

Next, there is a PID control system used to control the servo motor. The servo motor receives feedback from the yaw and pitch angles of the servo. Figure 2 shows the control system diagram. PID control has been employed to adjust the yaw and pitch angles of directional objects, achieving rapid setpoint convergence and low mean absolute errors (Wahyudi et al., 2021).

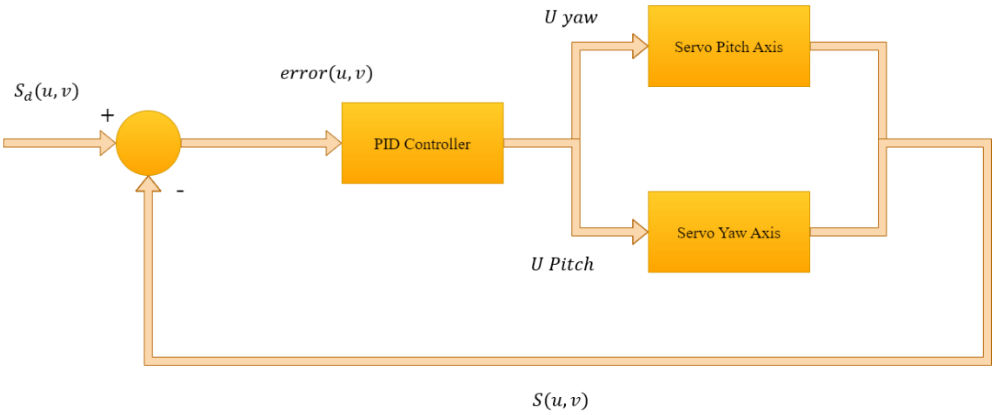

Figure 2. Control system design

Hardware Design

In this research, a microcontroller will connect various components as illustrated in Figure 3. A small 8MP camera handles image processing, while TD8120MG servo motors maneuver the camera along the yaw and pitch axes. The microcontroller will be an Arduino Uno, establishing communication with the laptop and other components through a serial connection. The image processing will be carried out on the laptop, ensuring that the system can effectively analyze and respond to visual data captured by the camera.

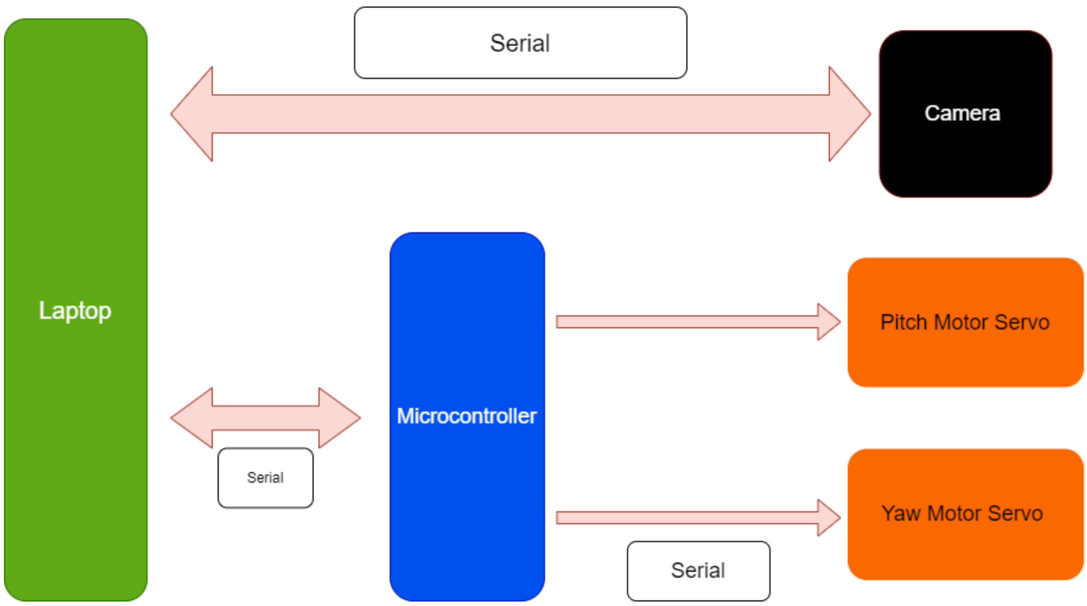

Figure 3. Hardware design

System Architecture

In this research, a procedural flowchart was created to ensure the research steps were correct and sequential, as shown in Figure 4. The research process is divided into three main tracks: data collection and processing, user interface design, and hardware design. The research begins with a literature review to understand the project's context and theoretical foundation. It then progresses to the design phase, which includes three main components: deep learning, user interface, and hardware.

Data is gathered and annotated in the data collection track, followed by image pre-processing, which includes resizing, data augmentation, and image enhancement. The processed data is then trained and tested using deep learning algorithms, and the results are evaluated. An interface is created using MIT App Inventor for the user interface track. Once the interface is developed, the application is exported. The TD8120MG servo motor with a bracket and camera is designed in the hardware design track. Components such as the Arduino Uno and servo yaw-pitch are integrated to build the necessary hardware.

Finally, all components are integrated to test the overall system functionality. This functional test aims to verify that the system operates as planned and that the measurements are accurate. The process concludes with writing the research report.

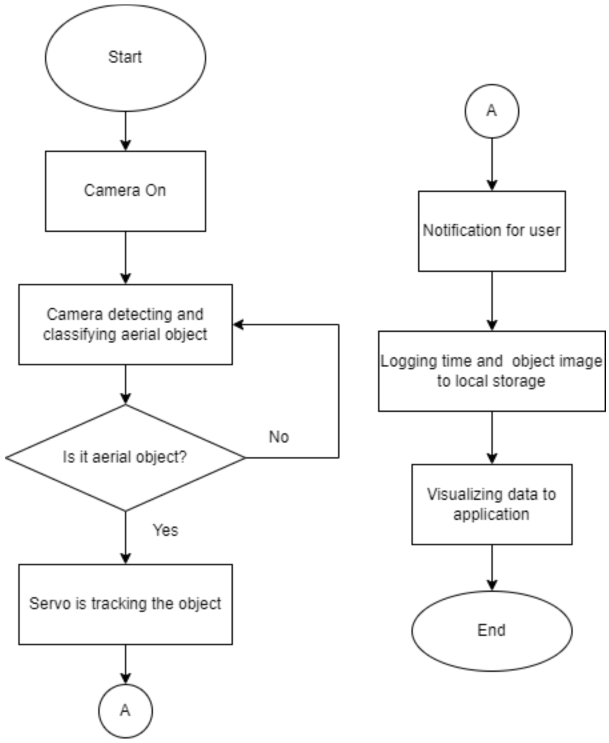

Figure 4. Step prototyping system flowchart

Deep Learning Model

In this research, there are several steps to modeling Deep learning as illustrated in Figure 5. In this research, 2,049 images of flying vehicles, specifically unmanned aircraft, were used for training using YOLOv8. The data used in this research is secondary data sourced from a dataset published on public.roboflow.com. The dataset was manually sorted by the researchers to include only images of unmanned aircraft in flight, excluding book covers, posters, and similar items. The author applied augmentation methods to create variations in the images, such as adding salt and pepper noise and altering brightness, increasing the total number of images to 7,227. Additionally, 652 images were used for validation to evaluate the model during training, ensuring better performance when deployed. Thus, the dataset was composed of 92% training data and 8% validation data.

Figure 5. Step modelling deep learning using YOLOv8

Each image processed for the detection algorithm was labeled with a classification code and the coordinates of the bounding box. Each line in the label file represents one object, with additional lines added for images containing multiple objects. Google Colab's V100 GPU was utilized for computing the training data, allowing faster training and greater flexibility in evaluating model performance. The YOLOv8 architecture was used to create the model, with 100 epochs for processing the data, resulting in the trained weights. When tested, the model demonstrated an accuracy of 77% with a threshold of 40% to minimize detection bias. These results can be further used for servo tracking.

Integration IoT with MIT App Inventor

This research implemented a real-time monitoring system using Flask to upload object detection to an Internet Protocol (IP) address. Flask is a Python-based microframework that simplifies web application development (Coburn, 2020; Relan, 2019). This IP address can access a local computer for streaming detected objects, but it can only be accessed by users on the same network as the local computer.

There are several steps to design flask python as shown in Figure 6. First, the web application was initialized using Flask. Flask is designed to simplify the creation of web applications by providing essential tools. The research involved setting up HTTP response handling for specific URLs. By defining these routes, the Flask application can display web pages rendered from HTML templates when users visit the site's homepage. This approach facilitates the development of dynamic and well-structured web applications, where presentation components and business logic can be managed separately but work together to provide a consistent and responsive user experience.

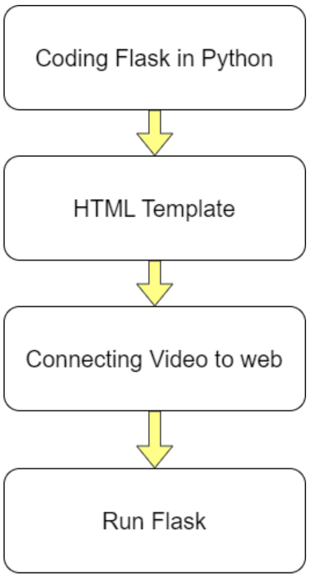

Figure 6. Step designing Flask Python

Researcher use function to sends a response containing video streaming to the web. This code configures the web application to stream video in real-time using HTTP techniques that allow partial replacement of already loaded content. Finally, the application is ensured to run as a web server when executed as the main script. This setup allows the Flask server to run and be accessible across the network, ensuring the web application is ready to receive and respond to client requests.

An interface was created using MIT App Inventor for smartphone users and also the step shown in Figure 7. The interface includes a label for the title, a web viewer to connect to a specific network, a button to initiate the connection after entering the URL in the textbox, and a textbox for inputting the destination URL. Additionally, a block diagram was designed to serve as the programming code for the interface, illustrated in Figure 4.27. It features a drag-and-drop interface and a blocks-based programming language, allowing users to focus on app logic rather than syntax. The block diagram shows that pressing the button triggers an action to connect the interface to the network specified in the textbox, and the URL is then displayed in the web viewer.

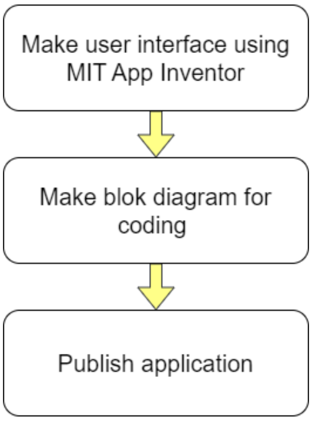

Figure 7. Step designing MIT App Inventor

Integration IoT with Telegram Notification

In this research, the notification system uses Telegram as the notification interface and the step shown in Figure 8. A bot was created to send detected object alerts, including text messages, images, and videos of detected events to operators (Bakri, 2021). A group was also set up with several members so multiple people can receive notifications in one chat group through internet (Sholahuddin et al., 2023). Users receive notifications with the time the object was detected, formatted as hours, minutes, and seconds (Renuka et al., 2023). The system is configured to send notifications to Telegram upon the first detection of an object and every 500 detections after that to prevent lagging. The function checks each detection result for detected objects by examining the length of the result data. If a detection is found, a global counter is incremented, and the current time is formatted as a string to be used as the image file name. The image is saved if it is the first processed image or every 500th image. The saved image is then opened in binary mode and sent to the Telegram bot using the `sendPhoto` method, with the detection time included as a caption. This process ensures that only images meeting specific conditions are saved and sent, allowing efficient and automated monitoring of detected objects.

Figure 8. Step Designing Telegram notification

RESULTS AND DISCUSSION

Prototype Design

In this research, the hardware system was designed using two servos, a bracket, and a small camera for detection, as shown in Figure 9. One servo controls the yaw, and the other controls the pitch on the bracket. Each servo is connected to an Arduino Uno, with the yaw servo connected to digital pin 9 and the pitch servo connected to digital pin 10. The camera is mounted on the bracket above the pitch servo and connected to a laptop for object detection computation.

Servo Response

The angle values of two servo motors using PID control over time are displayed in Figure 10, both with and without dynamic adjustment. This experiment was conducted to understand the long-term impact of servo angles on the object. The vertical axis shows the servo positions in degrees, ranging from 0 to 180, while the horizontal axis shows time in seconds. The figure features four lines: the red line for yaw and the orange line for pitch represent angles using PID control, and the dark blue line for yaw and light blue line for pitch represent angles without PID control. The graph shows that PID-controlled and non-PID-controlled servo angles follow similar trends but with significant differences. Servo angles using PID are relatively more stable, whereas those without PID fluctuate more drastically. This difference impacts servo performance during object detection. Additionally, servo angles with PID do not exceed 180 degrees or fall below 0 degrees, whereas, without PID, the pitch angle drops below 0 degrees, causing errors that stop the system since servos only operate within the 0 to 180-degree range.

Figure 9. Hardware prototype

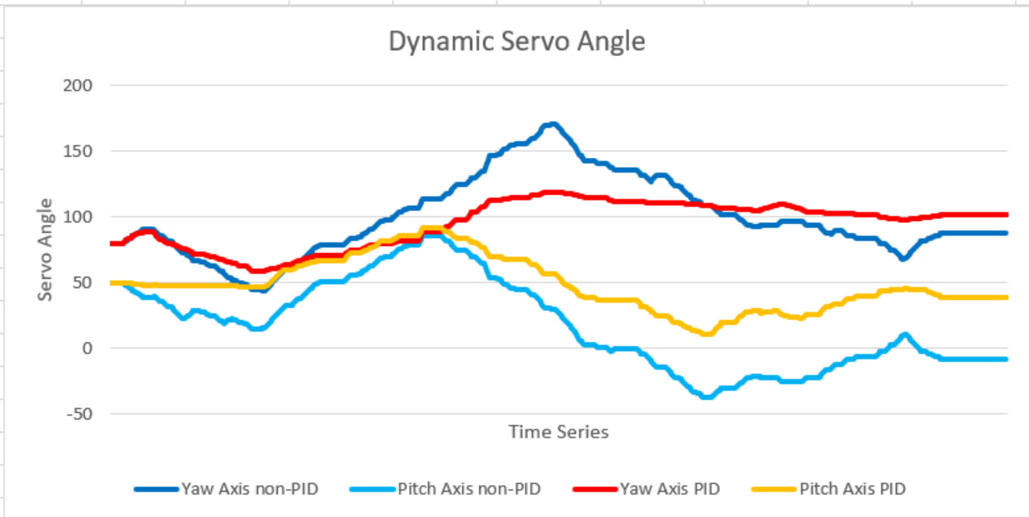

Figure 10. Comparison Graph of Servo Angles with PID and Not Using PID Dynamically

Figure 11 shows the angle values using the same method but with a static object and setpoints of 80 degrees for yaw and 50 degrees for pitch. This experiment was conducted to examine in detail the causes of the significant deviations observed in the previous dynamic experiment. The light blue and blue lines without PID exhibit a sawtooth pattern with varying values, while the red and orange lines with PID form more stable, straight lines. Using PID results in more precise and accurate angle values, significantly enhancing the system's performance in tracking objects.

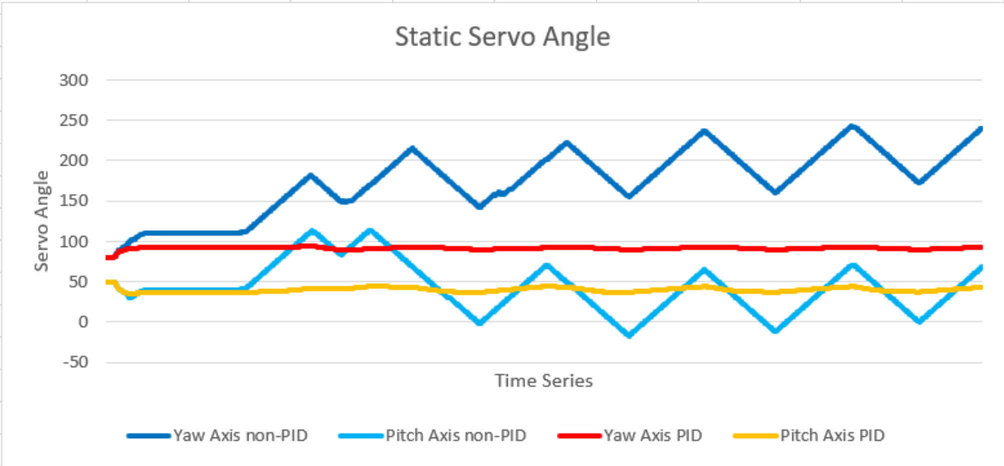

Figure 11. Comparison Graph of Servo Angles with PID and Not Using PID Statically

Deep Learning Model Evaluation

The In this research, four metrics were used to evaluate the performance of the object detection model: precision, recall, mAP50, and mAP50-95 as shown in Figure 12. Precision measures the accuracy of the model's positive predictions, with a high value indicating that most predicted objects are correct; this model has a precision of 86.2%. Recall measures the model's ability to find all positive cases, ensuring it does not miss objects it should detect; the model has a recall of 72.6%. The lower-left graph in Figure 9 shows the mean Average Precision (mAP) at an Intersection over Union (IoU) of 0.50, which measures how well the predicted bounding boxes match the actual ones, with a score of 78.1%. The mAP50-95 graph shows the mean Average Precision across a range of IoU thresholds from 0.50 to 0.95, scoring 42.4%. This metric averages the model's precision at various IoU levels, indicating that while the model performs well at different accuracy levels, there is more variability than using only IoU=0.50.

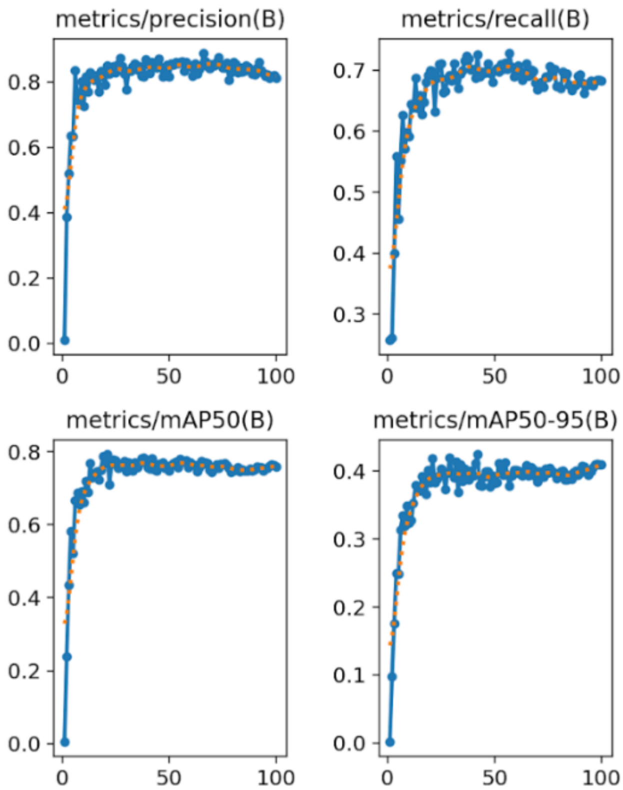

Figure 12. Results metrics YOLOv8

A normalized confusion matrix is shown to assess the model's performance as displayed in Figure 13. This matrix compares two classes: unmanned aircraft (drones) and background. The top left section, True Positive, has a value of 0.77, indicating that the model correctly predicted the "drone" class 77% of the time. The top right section, False Negative, has a value of 1, meaning the model never incorrectly predicted a "drone" as "background." The bottom left section, False Positive, shows a value of 0.23, indicating that 23% of the predictions for the "drone" class were actually "background." This error means the model mistakenly detected a drone when there was none in the frame. Finally, the bottom right section, True Negative, has a value of 1, showing that the model always correctly predicted the "background" class, meaning it never failed to detect a drone when one was present. Integration IoT with MIT Inventor App.

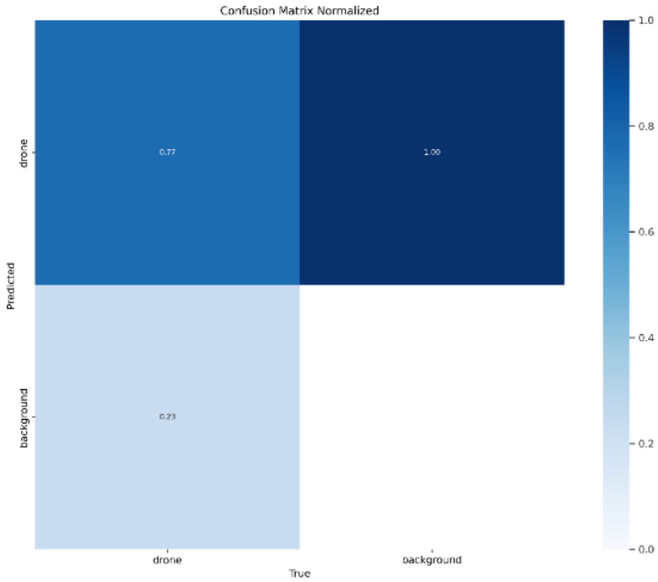

Figure 13. Model Confussion Matrix

Integration IoT with MIT App Inventor and Flask Python FPS

FPS is a crucial indicator of an application's visual smoothness and responsiveness, particularly in video and interactive applications. Figure 14 compares the real-time performance of Frames per Second (FPS) from three different sources. The web-based application shows an average FPS of 6.8. The locally run application displays a higher average FPS of 11.7, indicating better performance with nearly 12 frames updated per second. Suggests that local applications handle graphics processing more efficiently without the additional load from the browser. The application from MIT App Inventor has an average FPS of 7.6, performing better than the web-based version but still less efficiently than the locally run application.

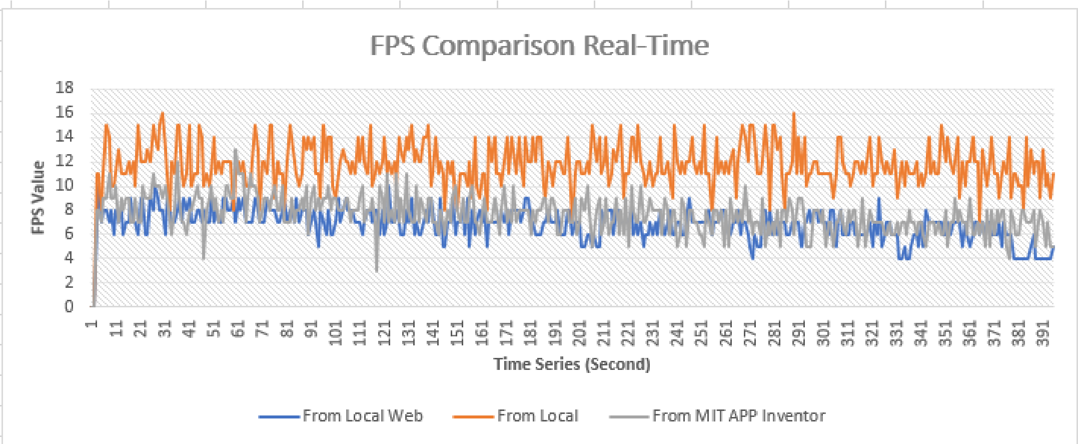

Figure 14. FPS Comparison chart using local network, local, and MIT App Inventor

Integration IoT with Telegram Notification FPS

The performance of frames per second (FPS) on the system was analyzed about the notifications received on Telegram over a period measured in seconds. The graph shows two categories: FPS performance with Telegram notifications and without Telegram notifications. Figure 15 illustrates that Telegram notifications affect system performance, reducing the average FPS to 9.6. Additionally, system performance with Telegram notifications shows greater fluctuations than without notifications, with an average FPS of 22.4. These fluctuations suggest the device's performance is more inconsistent when interrupted by notifications. The graph also indicates several instances where FPS drops to zero. This significant FPS drop occurs because sending notifications to Telegram requires high computational resources, leading to a marked decrease in FPS.

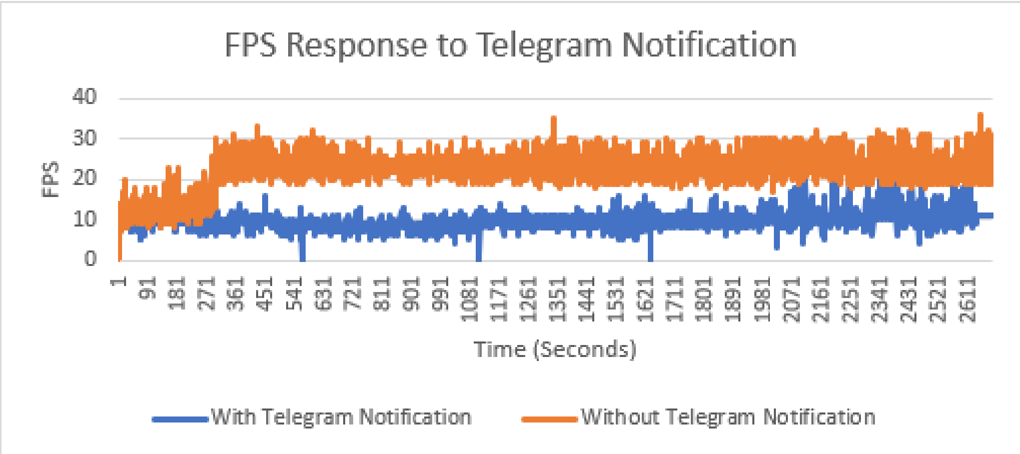

Figure 15. Graph of FPS Response to Telegram Notification

CONCLUSION

This research concentrates on developing a system for detecting, classifying, and reconnaissance flying objects using Convolutional Neural Network (CNN) and Internet of Military Things (IoMT) technology. This research takes an integrative approach by combining Doppler radar technology for motion detection and cameras for image processing, utilizing YOLOv8 as an architecture for object detection with an accuracy of 78.1%. Additionally, PID control is used for servo motor adjustment, ensuring dynamic object spotting. Then there is a notification via Telegram which will send a notification every time the camera detects an object. This research also developed a user interface using MIT App Inventor, which allows real-time monitoring and visualization via smartphone applications with FPS reaching 7.6. the system can also visualize via the local web with FPS reaching 6.8.

In this research, several suggestions can be made to enhance the effectiveness of the developed system for classifying and monitoring unmanned aerial vehicles (UAVs). Integrating Identification Friend or Foe (IFF) technology into the system is crucial. IFF will enable the system to differentiate between friendly and hostile UAVs. Using more advanced radar systems will improve initial detection and tracking accuracy, especially under poor visibility conditions or with fast-moving objects. Modern radar technology provides more precise data for image processing systems, thereby enhancing the overall system response. Additionally, gathering a larger and more diverse set of image data will aid in developing a more robust Convolutional Neural Network (CNN) model. A more representative dataset will help train the YOLOv8 algorithm to be more resilient across various operational scenarios. By incorporating a large and diverse dataset, the model can improve its learning capability, resulting in more accurate classifications and increased confidence in target identification. Implementing these suggestions is expected to contribute significantly to enhancing national security and the efficiency of air defense systems.

AUTHOR CONTRIBUTIONS

M.A.V.L.: Conceptualization, methodology, data curation, formal analysis, & writing – original manuscript. I.A.D.: Conceptualization, supervision, validation, & writing – review & editing. H.T.:

Supervision, validation, & resources. S.W.P.: Formal analysis, software, & validation. M.A.F.: Methodology & visualization.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

Andréfouët, S., Paul, M., & Farhan, A. R. (2022). Indonesia's 13558 islands: A new census from space and a first step towards a One Map for Small Islands Policy. *Marine Policy*, 135, 104848. https://doi.org/10.1016/j.marpol.2021.104848

Bakri, A. R. (2021). Implementasi intrusion detection system(ids)menggunakan telegram sebagai media notifikasi. *Jurnal Informatika dan Sistem Informasi*. https://api.semanticscholar.org/CorpusID:249899656

Bing, L., Gu, Y., Hu, L., Bowen, L., Lihua, Y., Wang, J., & Yin, Y. (2023). Ultra reliability and massive connectivity provision in integrated internet of military things (IoMT) based on tactical datalink. *Defence Technology*, 33, 386-398. https://doi.org/10.1016/j.dt.2023.09.016

Coburn, J. (2020). *Getting Started with Flask. In Build Your Own Car Dashboard with a Raspberry Pi* (pp. 143–170). Apress. https://doi.org/10.1007/978-1-4842-6080-7_6

Fahriannur, A., & Siswanto, M. (2017). Sistem Tracking Obyek Berbasis Algoritma Optical Flow Menggunakan Kamera Pan-Tilt. *Seminar Nasional Hasil Penelitian Tahun 2017.* https://publikasi.polije.ac.id/index.php/prosiding/article/view/780

Huffaker, J. D., Barker, C. K., Brown, D. M., Chesley, D. E., Clark, T. R., Dockery, D., Farthing, C. L., Griffith, K. A., Ho, M. T., Hutton, S. T., Juarez, J. C., Krichene, H., Mule, J. N., O'Haver, K. W., Phillips, S. A., Ricciardi, G. F., & Talisa, S. H. (2020). Sensors and Communications Systems. *Johns Hopkins APL Technical Digest,* 35(2), 123-146. https://api.semanticscholar.org/CorpusID:220126715

Ismail, M. S., Ahmad, A., Ismail, S., & Yusop, N. M. M. (2022). A review on Unmanned Aerial Vehicle (UAV) threats assessments. *AIP Conference Proceedings,* 2617(2022), 050007. https://doi.org/10.1063/5.0119703

Khan, F. R., Muhabullah, Md., Islam, R., Monirujjaman Khan, M., Masud, M., Aljahdali, S., Kaur, A., & Singh, P. (2021). A Cost-Efficient Autonomous Air Defense System for National Security. *Security and Communication Networks*, 2021, 1–10. https://doi.org/10.1155/2021/9984453

Khansa, A., Hemas, K. D., & Moses, G. R. P. (2021). Archipelago Insights as Efforts to Establish National Geostrategic Resilience: A Literature Review. *Preprints 2021,* 2021050154*.* https://doi.org/10.20944/preprints202105.0154.v1

Khawaja, W. A. G., Ezuma, M., Semkin, V., Erden, F., Ozdemir, O., & Guvenc, I. (2022). A Survey on Detection, Tracking, and Classification of Aerial Threats using Radars and Communications Systems. https://api.semanticscholar.org/CorpusID:253708184

Nosach, A. V. (2019). Factor system for the determination of offences against the sovereignty and territorial integrity of Ukraine. *Uzhhorod National University Herald. Series: Law*, 58(2), 110–115. https://doi.org/10.32782/2307-3322.58-2.24

Paliz Ochoa, P. X., Acosta Bedon, J. F., Tiuma, A., & Bravo, M. (2022). Advances in air defense systems. *Athenea*, 15–25. https://doi.org/10.47460/athenea.v3i9.42

Piekarski, M., & Wojtasik, K. (2022). Protection of Polish critical infrastructure (CI) against air threats. *Security and Defence Quarterly*. https://doi.org/10.35467/sdq/147676

Relan, K. (2019). *Beginning with Flask. In Building REST APIs with Flask* (pp. 1–26). Apress. https://doi.org/10.1007/978-1-4842-5022-8_1

Renuka, N., Panneer Selvam, M., Ponmurugan, P., Kiruthika, S., Selvam, N., & Kannan, N. (2023). Automated Email Notifications and Security Monitoring for Cloud Storage Access using Cloud Function. *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA),* 1205–1210. https://doi.org/10.1109/ICIMIA60377.2023.10425959

Rosłan, G. (2023). A description of military threats to poland's air safety. *Humanities and Social Sciences Quarterly*, 30(3), 117–126. https://doi.org/10.7862/rz.2023.hss.31

Saptono, E., Sudarya, A., & Pudjiatmoko, S. (2022). The synergy of human resources management between the regional governments of the Riau Islands Province with elements of defense in facing nonmilitary threats. *Technium Social Sciences Journal*, 27, 546–555. https://doi.org/10.47577/tssj.v27i1.5426

Sholahuddin, M. R., Atqiya, F., Wulan, S. R., Harika, M., Fitriani, S., & Sofyan, Y. (2023). Implementasi Sistem Identifikasi Senjata Real Time Menggunakan YOLOv7 dan Notifikasi Chat Telegram. *Journal of Information System Research (JOSH)*, 4(2), 598–606. https://doi.org/10.47065/josh.v4i2.2774

Sunardi, Widodo, A. M., Putra, K. T., Darmarjati, C., & Prayitno. (2022). Design of IoT-Based Control System Using Fuzzy Inference System. *Jurnal Polimesin*, 20(2). http://dx.doi.org/10.30811/jpl.v20i2.2650

Wahyudi, W., Saputra, B., & Syafei, W. (2021). Drone Tracking System Using GPS and Barometer Sensor. *Proceedings of the 2nd International Conference on Industrial and Technology and Information Design, ICITID 2021*, 30 August 2021, Yogyakarta, Indonesia. https://doi.org/10.4108/eai.30-8-2021.2311519

Zhang, D., Prabhakara, A., Munir, S., Sankaranarayanan, A. C., & Kumar, S. (2021). A Hybrid mmWave and Camera System for Long-Range Depth Imaging. *ArXiv*, abs/2106.07856. https://api.semanticscholar.org/CorpusID:235435669