Jurnal Otomasi Kontrol dan Instrumentasi Vol18 (2), 2026 ISSN: 2085-2517
https://doi.org/10.5614/joki.2026.18.2.14 E-ISSN: 2460-6340

Effect of Camera Type on Drone Detection Performance and UAV Control Response Based on YOLOv8 and YOLOv11

Pengaruh Jenis Kamera terhadap Performa Deteksi Drone dan Respons Kendali UAV Berbasis YOLOv8 dan YOLOv11

[1]Nesha Roswanda Sakhdiah, [1]Bagus Rangge Plaewe Nusantara,[2]Handoko,[2]Giri Yudho,[1]Khazali Fahmi*)

[1] Universitas Pertahanan Republik Indonesia, Bogor 16810, Indonesia

[2] Dinas Penelitian dan Pengembangan TNI AU, Indonesia

*) *corresponding email: fahmikhazali1@gmail.com*

Abstract

The development of Unmanned Aerial Vehicle (UAV) technology requires drone detection systems that are accurate, fast, and capable of operating in real time. This study analyzes the effect of camera characteristics on drone detection performance and UAV control response using YOLOv8 and YOLOv11. Experimental testing was conducted using five cameras with 720p and 1080p resolutions at distances of 1–5 m, with real-time inference performed using a MacBook Air M1. Distance estimation was performed using a monocular distance estimation approach based on the drone's bounding box height, object height, and camera focal length. The evaluation parameters included detected distance, MAE, RMSE, FPS, roll and pitch control values, and left and right elevon angles. The results show that camera characteristics affect distance estimation accuracy, detection speed, and UAV control output. YOLOv11 with Logitech BCC950 achieved an MAE of 0.34 m and RMSE of 0.39 m, but had low FPS. Logitech HD720P achieved 15.8 FPS with an MAE of 0.38 m and RMSE of 0.43 m.

Keywords: UAV, drone detection, YOLOv8, YOLOv11, distance estimation.

Abstrak

Perkembangan teknologi Unmanned Aerial Vehicle *(UAV) menuntut sistem deteksi* drone *yang akurat, cepat, dan mampu bekerja secara* real time*. Penelitian ini menganalisis pengaruh karakteristik kamera terhadap performa deteksi* drone *dan respons kendali UAV menggunakan YOLOv8 dan YOLOv11. Pengujian dilakukan secara eksperimental menggunakan lima kamera beresolusi 720p dan 1080p pada jarak 1–5 m dengan inferensi real time menggunakan MacBook Air M1. Estimasi jarak dilakukan menggunakan pendekatan* monocular distance estimation *berdasarkan tinggi* bounding box *drone, tinggi objek, dan panjang fokus kamera. Parameter evaluasi meliputi jarak terdeteksi, MAE, RMSE, FPS, nilai kendali* roll *dan* pitch*, serta sudut elevon kiri dan kanan. Hasil menunjukkan bahwa karakteristik kamera memengaruhi akurasi estimasi jarak, kecepatan deteksi, dan keluaran kendali UAV. YOLOv11 dengan Logitech BCC950 menghasilkan MAE 0.34 m dan RMSE 0.39 m, tetapi FPS rendah. Logitech HD720P menghasilkan FPS 15.8 dengan MAE 0.38 m dan RMSE 0.43 m.*

Kata Kunci: UAV, deteksi drone, YOLOv8, YOLOv11.

Paper accepted Augustus 5th, 2026 – paper revised September 21th, 2026 – approved October 19th, 2026

This paper is open access with CC BY-SA *license*.

1 Introduction

With the rapid development of technology and advances in electronic devices, Unmanned Aerial Vehicle (UAV) technology has been increasingly applied in environmental monitoring and risk assessment, including coastal and island surveillance applications [1]. The increasing use of UAVs not only provides various benefits but also raises security challenges associated with unidentified drones or unauthorized drone operations within a given area [2]. Therefore, a drone detection system capable of operating accurately and in real time is required to support surveillance and aerial target interception. However, conventional detection methods generally require high computational resources, involve longer processing times, and have limited feature extraction capabilities, making them less optimal for detecting UAV targets rapidly and accurately [3].

The development of computer vision and deep learning technologies has provided opportunities to improve the capability of automated drone detection systems [4]. One of the widely used algorithms for object detection is You Only Look Once (YOLO), which is known for its ability to perform real-time detection while maintaining good

accuracy [5]. Various developments within the YOLO family have been introduced to improve detection performance, processing speed, and computational efficiency [6]. The development of YOLO architectures across different versions is illustrated in Figure 1, which shows the evolution of YOLO models in supporting improvements in object detection capabilities.

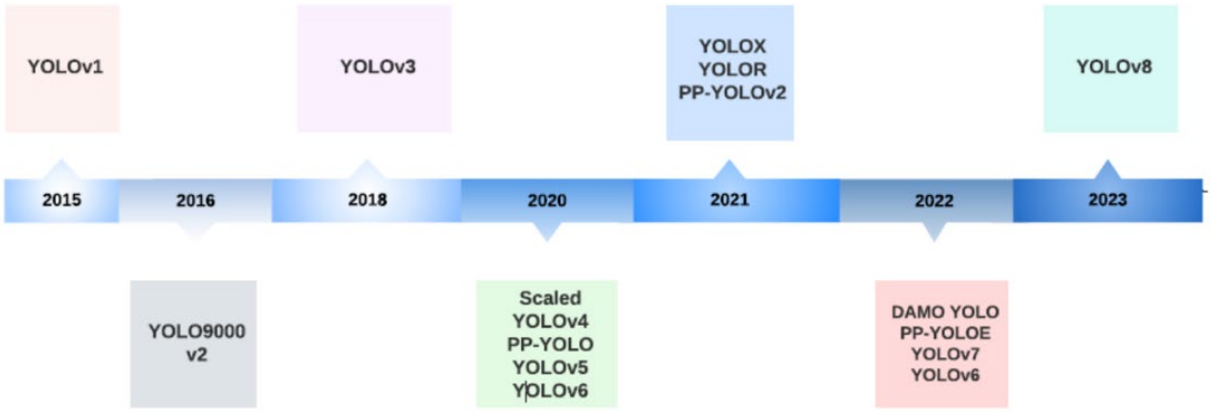


Figure 1. Evolution of YOLO architectures. Source: Yilmas, 2024.

In addition, optimization of YOLO architectures has been shown to improve UAV detection accuracy while reducing model complexity, making them more suitable for deployment on devices with limited computational resources without compromising real-time processing capabilities [7]. In drone detection applications, the system is required to recognize small targets that frequently appear against complex backgrounds. This condition presents a challenge because drones occupy relatively small regions of an image, resulting in limited feature information that can be extracted. Therefore, various studies have developed YOLO-based architectures to improve multi-scale feature extraction, strengthen semantic representation, and reduce detection errors without decreasing processing speed [8]–[10]. In addition to detection accuracy, Frame Per Second (FPS) is also an important parameter because it determines the ability of a system to process changes in the target position rapidly and continuously in real-time applications [8]–[10].

Although many studies have focused on developing YOLO architectures to improve detection accuracy and processing speed, the performance of a computer vision system is influenced not only by the detection algorithm but also by the quality of images produced by the camera device [11]. Differences in camera characteristics, such as resolution, sensor quality, object detail capture capability, and performance under varying lighting conditions, may affect the resulting detection performance. High-quality images allow object features to be extracted more effectively, whereas low-quality images may lead to reduced detection accuracy, particularly for small objects such as drones [12]. In UAV systems based on visual tracking, object detection results are not only used to identify targets but also serve as inputs to the flight control system. Target position and distance information obtained from the detection process can be used to generate automated control commands, enabling the UAV to adjust its direction and movement in real time [13].

Previous studies have demonstrated that detection and tracking results can be utilized as inputs to control systems to regulate UAV movement through pitch and roll parameters, as well as control-surface actuation, to maintain the target within the camera's field of view [14]. Previous research has also investigated the influence of camera-related and environmental factors on drone detection performance. Ayuningtyas et al. evaluated the effects of camera type, light intensity, camera-to-object distance, and drone altitude on a YOLOv5-based real-time drone surveillance system [15]. However, their study focused primarily on detection accuracy and did not examine how variations in camera characteristics propagate into downstream UAV control responses. In contrast, the present study compares five camera devices using YOLOv8 and YOLOv11 and evaluates their effects on distance estimation, processing speed, roll, pitch, and elevon control outputs.

Therefore, this study aims to evaluate the effect of camera type variations on distance estimation accuracy, detection speed, and elevon control response in a YOLOv8- and YOLOv11-based drone detection system. The evaluation is conducted using detected distance, FPS, pitch, roll, and the movement of the left and right elevons as performance parameters. The results are expected to provide information regarding the influence of camera characteristics on detection quality and UAV control response, as well as serve as a reference for selecting camera devices for real-time drone detection and tracking applications.

2 Method

This study employed an experimental method to analyze the effect of camera type variations on drone detection performance using the YOLOv8 and YOLOv11 algorithms. The experiment was conducted by comparing the drone detection results obtained from several camera types under identical testing conditions. The evaluation was based on the detected distance, Frame Per Second (FPS), pitch, roll, left elevon, and right elevon parameters generated during the target tracking process.

2.1 Research Equipment and Materials

The hardware used in this study consisted of a drone, a processing computer, an Arduino Nano as a microcontroller, two SG90 servo motors as actuators for the left and right elevons, and five different camera types used as image acquisition sources, as shown in Figure 2. The software used included the operating system, Python, OpenCV, Arduino IDE, and YOLOv8 and YOLOv11 object detection models trained using a drone dataset. The detection models were used to identify the position of the drone in the image in real time and generate data used as input for the control system.



(a) (b) (c) (d) (e)

Figure 2. Hardware used in the study: (a) iPhone 12 Pro, (b) Infinix Smart 20, (c) Logitech HD720P, (d) Logitech BCC950, and (e) MacBook Air M1.

Table 1. Technical specifications of the cameras used in the study.

Code	Camera	Type	Video Resolution
K1	iPhone 12Pro	Smartphone	1080p
K2	Macbook Air M1	Laptop camera	720p
K3	Infinix Smart 20	Smartphone	1080p
K4	Logitech BCC950	Webcam	1080p
K5	Logitech HD720P	Webcam	720p

As shown in Table 1, the cameras used in this study had two video resolution levels, namely 720p and 1080p. Camera codes were used because several devices had the same video resolution, allowing each device to be distinguished during the testing and result analysis stages.

2.2 System Architecture

The developed system consisted of several interconnected stages, including system initialization, image acquisition, image preprocessing, object detection, distance estimation, control response calculation, elevon actuation, and performance evaluation. At the initial stage, either the YOLOv8 or YOLOv11 model was loaded on the processing computer, while the selected camera was activated as the image input source. The camera then captured real-time video of the drone according to the resolution supported by each device. The acquired images were subsequently processed to match the input requirements of the detection model before object detection was performed.

During the object detection stage, YOLOv8 or YOLOv11 was used to identify the presence of the drone and generate information in the form of a bounding box and target position in the image plane. The height of the detected bounding box was then used in the distance estimation process using a monocular distance estimation method. The target distance was calculated based on the actual object height, the camera focal length, and the bounding box height in the image. Therefore, the output of the detection stage was used not

only to determine the presence of the target but also to estimate the distance between the camera and the drone.

In addition to distance information, the center position of the bounding box was used to determine the deviation of the target position from the center of the camera field of view. This positional information was subsequently processed by the control system to generate roll and pitch values as references for UAV motion correction. The control values were then transmitted from the processing computer to the Arduino Nano through serial communication. The Arduino Nano processed the received data and generated Pulse Width Modulation (PWM) signals to drive two SG90 servo motors representing the left and right elevon actuators. Changes in the angles of both elevons were used to simulate the UAV control response to variations in the detected target position.

Throughout the process, the system also recorded real-time evaluation parameters, including detected distance, FPS, roll and pitch values, and left and right elevon angles. The estimated distance was subsequently compared with the actual distance to obtain the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values. These parameters were used to evaluate the effect of camera type variations on distance estimation accuracy, processing speed, and UAV control response. The overall workflow and interactions among the system components are illustrated in Figure 3.

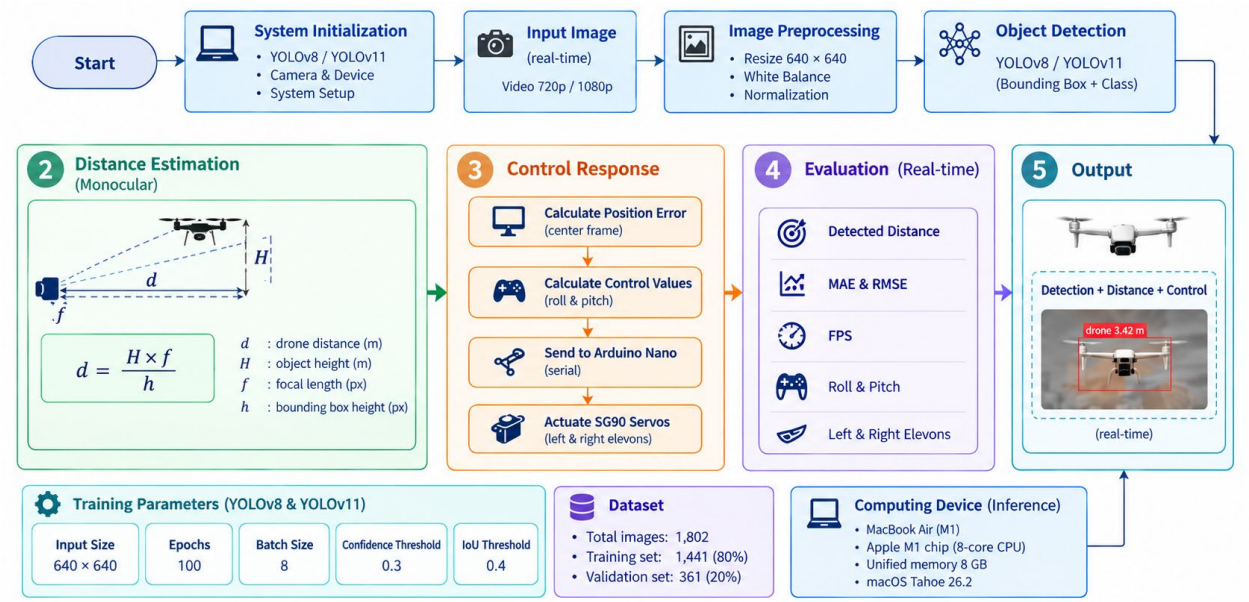


Figure 3. Architecture of the research system.

2.3 Research Variables

The research variables were divided into independent, dependent, and controlled variables. The independent variable in this study was the camera characteristics used as the source of the system input images. The dependent variables were the parameters observed to evaluate the effect of camera type variations on system performance. These parameters included UAV detection distance, FPS, pitch angle, roll angle, left elevon position, and right elevon position generated by the control system based on the object detection results. To ensure that differences in the experimental results were primarily attributable to the camera type, several variables were controlled and maintained throughout the experiment. These controlled variables included the drone model, testing environment, testing distances, processing computer specifications, Arduino Nano, SG90 servo motors, model training dataset, and YOLO inference parameters used during testing. Two object detection models, YOLOv8 and YOLOv11, were used in this study. Both models were trained using the same parameters to ensure an objective comparison. A dataset consisting of 1,802 images was randomly divided into 1,441 training images (80%) and 361 validation images (20%), without using a separate test set. All images used an input size of 640 × 640 pixels. The training process was conducted for 100 epochs with a batch size of 8, using the automatic optimizer (optimizer = auto) and an initial learning rate of 0.01. Training was performed using the CPU for both models with the same configuration. During inference, the confidence threshold and IoU threshold were set to 0.3 and 0.4, respectively, for both models. Inference and testing were performed using a MacBook Air (M1, 2020) equipped with an Apple M1 chip, 8 GB of unified memory, and macOS Tahoe 26.2. The same computing device was used for all YOLOv8 and YOLOv11 experiments to maintain consistent testing conditions, particularly for processing speed and FPS measurements.

2.4 Experimental Procedure

The experiment was conducted using five different camera types as image input sources. Each camera was tested to detect a target drone at distances of 1 m, 2 m, 3 m, 4 m, and 5 m. For each test, the camera was first connected to the processing computer running the YOLOv8- or YOLOv11-based object detection algorithm. The target drone was then positioned at the predetermined distance while the system performed real-time object detection and tracking.

The target position coordinates obtained from the detection results were subsequently processed to generate pitch and roll values as references for UAV movement. These values were transmitted to the Arduino Nano through serial communication and processed into actuator control signals. Based on the received signals, the Arduino Nano controlled two SG90 servo motors functioning as the left and right elevon actuators. During the testing process, data including detection success, detected distance, pitch, roll, elevon movement, and FPS were recorded and stored for further analysis.

To ensure the validity and objectivity of the results, all testing scenarios were conducted under identical conditions, including the testing environment, target position, hardware specifications, model inference parameters, and system configuration. Therefore, differences in the obtained results could be attributed primarily to differences in the characteristics of the tested cameras.

The control-response experiment in this study was conducted statically using a bench-test prototype rather than on a free-flying UAV. The detected target position was used to generate pitch and roll control values, which were then transmitted to the Arduino Nano to actuate the SG90 servos representing the left and right elevons. Therefore, the observed control response represents the actuator response to the computer-vision output under static test conditions. To improve data consistency, each test at distances of 1 m, 2 m, 3 m, 4 m, and 5 m was repeated twice for every camera–YOLO model combination. The resulting measurements were used to evaluate detected distance, FPS, control values, and elevon response.

2.5 Data Acquisition

The research data were obtained through UAV detection and tracking experiments using different camera types and object detection models. The system testing and data acquisition process is shown in Figure 4. In each experiment, the target drone was positioned at five different distances, namely 1 m, 2 m, 3 m, 4 m, and 5 m from the camera. During the detection process, the system automatically recorded several parameters used as research data.



Figure 4. Data acquisition process.

The recorded parameters included the actual target distance, distance detected by the system, FPS generated during the inference process, pitch and roll values obtained from the target position in the image plane, and the left and right elevon angles generated by the control system. All data were recorded for each combination of camera type and detection model. Thus, each camera–model combination generated five sets of test data representing system performance at distances ranging from 1 m to 5 m. The collected data were subsequently used to evaluate the effect of camera type on UAV detection capability, system processing speed, and the resulting elevon control response.

2.6 Data Analysis

Data analysis was performed quantitatively and descriptively to compare the performance of each camera when used with the YOLOv8 and YOLOv11 models.

2.6.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) was used to determine the average error in distance estimation. MAE was calculated using Equation (1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | J_a - J_d | \tag{1}$$

where J_a is the actual distance (m), J_d is the detected distance (m), and n is the number of test data points.

2.6.2 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) was used to evaluate the error in distance estimation by assigning greater weight to larger errors. RMSE was calculated using Equation (2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (J_a - J_d)^2} \tag{2}$$

In addition to distance estimation accuracy, FPS was used to evaluate the capability of the system to perform real-time detection. The pitch, roll, left elevon, and right elevon parameters were analyzed to determine the influence of camera variations on the UAV control response during target tracking. The analysis results were presented in tables and grap˙hs to facilitate comparisons between the different camera types and detection models.

2.6.3 Distance Estimation

Drone distance estimation was performed based on the relationship between the object size in the image and its distance from the camera using a monocular distance estimation method. The target distance was calculated from the actual object height, the effective camera focal length in pixels, and the height of the drone bounding box in the image, as expressed by equation 3.

$$d = \frac{H \times f}{h} \tag{3}$$

where d is the distance between the drone and the camera (m), H is the actual height of the drone (m), f is the camera focal length in pixels, and (h) is the height height of the detected drone bounding box in the image (pixels). The value of (h) was obtained from the detection results of YOLOv8 and YOLOv11 and was subsequently used to calculate the detected distance at each testing distance.

Because the cameras used in this study have different sensor characteristics, lenses, resolutions, and fields of view, the focal length in pixel units was not assumed to be identical for all devices. Therefore, each camera was calibrated independently prior to the main experiments. A drone with a known actual height of (H=0.15) m was positioned at a known calibration distance, and the corresponding bounding-box height h_{cal} was obtained from the detection result. The effective focal length of each camera was calculated in equation 4.

$$f = \frac{d_{cal} \times h_{cal}}{H} \tag{4}$$

where d_{cal} represents the known actual distance during calibration. The resulting focal-length value for each camera was subsequently kept constant throughout all experiments performed with that particular camera. This approach allowed differences in the optical characteristics of the camera devices to be considered in the distance estimation process. Following calibration, the bounding-box height h obtained from YOLOv8 and YOLOv11 detection was combined with the corresponding calibrated focal length of each camera to estimate the target distance. The estimated distances were then compared with the actual distances to calculate the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

3 Results and Discussion

3.1 Drone Distance Estimation Test Results

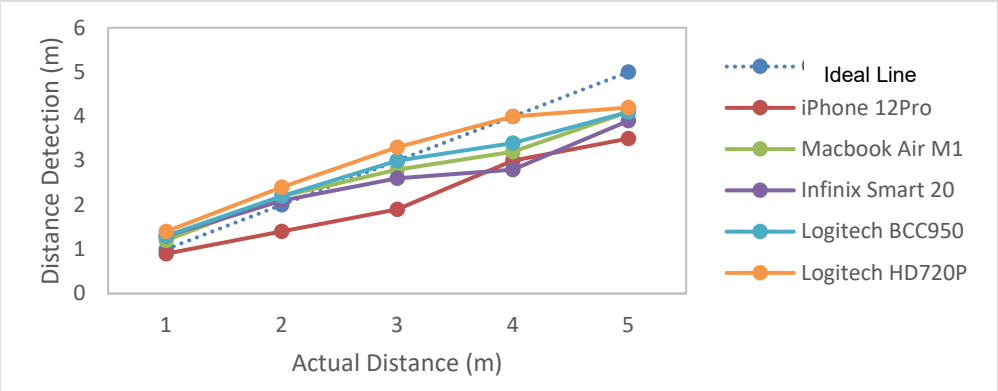
The experiments were conducted to investigate the effect of camera type variations on drone detection performance using the YOLOv8 and YOLOv11 models. The tests were performed at five actual distances, namely 1 m, 2 m, 3 m, 4 m, and 5 m. The main parameters observed included detected distance, FPS, roll, pitch, and left and right elevon angles as outputs of the control system. A summary of the distance estimation results and average FPS is presented in Table 2.

Table 2. Distance estimation results and average FPS for each camera–YOLO model combination.

Model	Code	Camera	Jd 1 m	Jd 2 m	Jd 3 m	Jd 4 m	Jd 5 m	Average FPS
YOLOv8	K1	iPhone 12 Pro	0.9	1.4	1.9	3	3.5	15.6
YOLOv11			1.5	1.8	2.4	2.8	2.9	15.4
YOLOv8	K2	Macbook Air M1	1.2	2.2	2.8	3.2	4.1	15.8
YOLOv11			1.4	2.2	2.9	3.5	3.9	15.2
YOLOv8	K3	Infinix Smart 20	1.3	2.1	2.6	2.8	3.9	8
YOLOv11			1.8	3.1	4	4.6	4.9	8.2
YOLOv8	K4	Logitech BCC950	1.3	2.2	3	3.4	4.1	5
YOLOv11			1.1	2.3	3.3	3.7	4.3	4.8
YOLOv8	K5	Logitech HD720P	1.4	2.4	3.3	4	4.2	16
YOLOv11			1.5	2.2	2.9	3.6	4.3	15.8

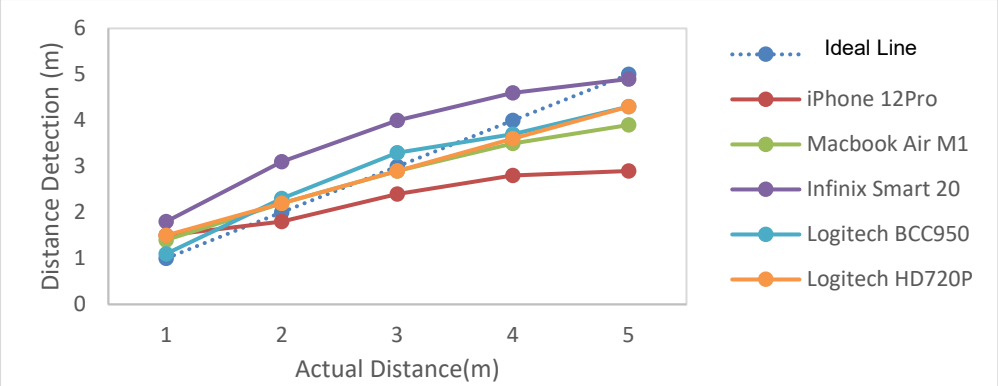
The detected-distance results relative to the actual distances show that each camera produced different distance estimation characteristics. In the YOLOv8 tests, cameras K5 and K4 exhibited estimation patterns that were relatively close to the ideal line compared with the other cameras. K5 produced detected distances of 1.4 m, 2.4 m, 3.3 m, 4.0 m, and 4.2 m at actual distances of 1–5 m, respectively, while K4 produced detected distances of 1.3 m, 2.2 m, 3.0 m, 3.4 m, and 4.1 m.

In the YOLOv11 tests, K4 produced detected distances that were relatively close to the actual distances, namely 1.1 m, 2.3 m, 3.3 m, 3.7 m, and 4.3 m at actual distances of 1–5 m, respectively. Meanwhile, K1 tended to produce lower distance estimates than the actual distances, particularly at 4–5 m. This indicates a tendency toward underestimation for this camera. These differences in estimation patterns indicate that camera characteristics may affect the quality of the images received by the detection model and consequently influence the resulting drone distance estimation.



(a)

Figure 5. Comparison between actual and detected distances for different camera types using (A) YOLOv8 and (B) YOLOv11.



(b)

Figure 5. (Cont.) Comparison between actual and detected distances for different camera types using (A) YOLOv8 and (B) YOLOv11.

3.2 Distance Estimation Error Analysis Based on MAE and RMSE

Distance estimation performance was evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE represents the average magnitude of the absolute error between the actual and detected distances. Meanwhile, RMSE assigns greater weight to larger errors and can therefore indicate the effect of relatively large deviations in the distance estimation results. Lower MAE and RMSE values indicate smaller distance estimation errors. The calculated MAE, RMSE, and average FPS values are presented in Table 3.

Table 3. MAE, RMSE, and average FPS values.

Model	Code	Camera	MAE (m)	RMSE (m)	Avarage FPS
YOLOv8	K1	iPhone 12 Pro	0.86	0.98	15.6
YOLOv11			0.92	1.14	15.4
YOLOv8	K2	Macbook Air M1	0.45	0.56	15.8
YOLOv11			0.46	0.57	15.2
YOLOv8	K3	Infinix Smart 20	0.62	0.76	8
YOLOv11			0.72	0.8	8.2
YOLOv8	K4	Logitech BCC950	0.4	0.5	5
YOLOv11			0.34	0.39	4.8
YOLOv8	K5	Logitech HD720P	0.38	0.45	16
YOLOv11			0.38	0.43	15.8

The K4–YOLOv11 combination produced the lowest MAE and RMSE among the tested combinations, with values of 0.34 m and 0.39 m, respectively. These values indicate the smallest average absolute error and root mean square error in the tested dataset. The K5–YOLOv11 combination also produced relatively small errors, with an MAE of 0.38 m and an RMSE of 0.43 m. These values were only slightly higher than those obtained with K4 and were lower than the corresponding values obtained with K1, K2, and K3 using YOLOv11. These results indicate that a higher FPS does not necessarily correspond to smaller distance estimation errors.

3.3 FPS Analysis and Real-Time Processing Capability

In addition to distance estimation performance, FPS was used to evaluate the capability of the system to perform real-time detection. Based on the results presented in Figure 6, K5 produced the highest average FPS, reaching 16 FPS with YOLOv8 and 15.8 FPS with YOLOv11. K2 and K1 also produced relatively high FPS values of approximately 15 FPS. In contrast, K3 produced an average FPS of approximately 8 FPS, while K4 produced the lowest FPS values, namely 5 FPS with YOLOv8 and 4.8 FPS with YOLOv11. When compared with the distance estimation results, the increase in FPS did not exhibit a directly proportional relationship with improved distance estimation performance. For example, with YOLOv11, K4 produced the lowest FPS of 4.8

FPS but also resulted in the lowest MAE and RMSE values, namely 0.34 m and 0.39 m, respectively. Meanwhile, K5 produced the highest FPS of 15.8 FPS with an MAE of 0.38 m and an RMSE of 0.43 m. K1, which also had a relatively high FPS of 15.4 FPS, produced an MAE of 0.92 m and an RMSE of 1.14 m. These results indicate that FPS and distance estimation error represent different aspects of system performance and do not necessarily improve simultaneously. FPS represents the image processing speed, whereas MAE and RMSE represent the magnitude of distance estimation error. Therefore, the selection of a camera and detection model combination for real-time applications should consider the balance between processing speed and distance estimation performance.

The developed system consisted of several interconnected stages, including system initialization, image acquisition, image preprocessing, object detection, distance estimation, control response calculation, elevon actuation, and performance evaluation. At the initial stage, either the YOLOv8 or YOLOv11 model was loaded on the processing computer, while the selected camera was activated as the image input source. The camera then captured real-time video of the drone according to the resolution supported by each device. The acquired images were subsequently processed to match the input requirements of the detection model before object detection was performed. During the object detection stage, YOLOv8 or YOLOv11 was used to identify the presence of the drone and generate information in the form of a bounding box and target position in the image plane. The height of the detected bounding box was then used in the distance estimation process using a monocular distance estimation method. The target distance was calculated based on the actual object height, the camera focal length, and the bounding box height in the image. Therefore, the output of the detection stage was used not only to determine the presence of the target but also to estimate the distance between the camera and the drone. In addition to distance information, the center position of the bounding box was used to determine the deviation of the target position from the center of the camera field of view. This positional information was subsequently processed by the control system to generate roll and pitch values as references for UAV motion correction. The control values were then transmitted from the processing computer to the Arduino Nano through serial communication. The Arduino Nano processed the received data and generated Pulse Width Modulation (PWM) signals to drive two SG90 servo motors representing the left and right elevon actuators. Changes in the angles of both elevons were used to simulate the UAV control response to variations in the detected target position. Throughout the process, the system also recorded real-time evaluation parameters, including detected distance, FPS, roll and pitch values, and left and right elevon angles. The estimated distance was subsequently compared with the actual distance to obtain the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values. These parameters were used to evaluate the effect of camera type variations on distance estimation accuracy, processing speed, and UAV control response. The overall workflow and interactions among the system components are illustrated in Figure 3.

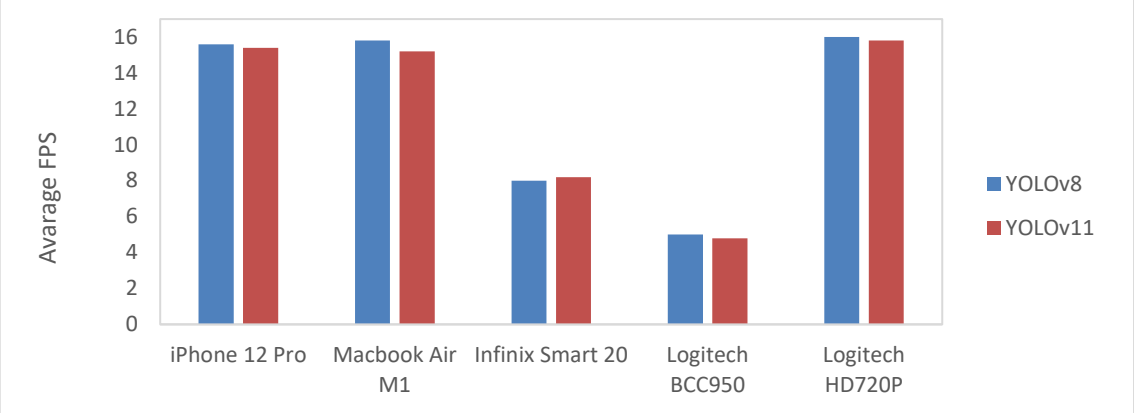
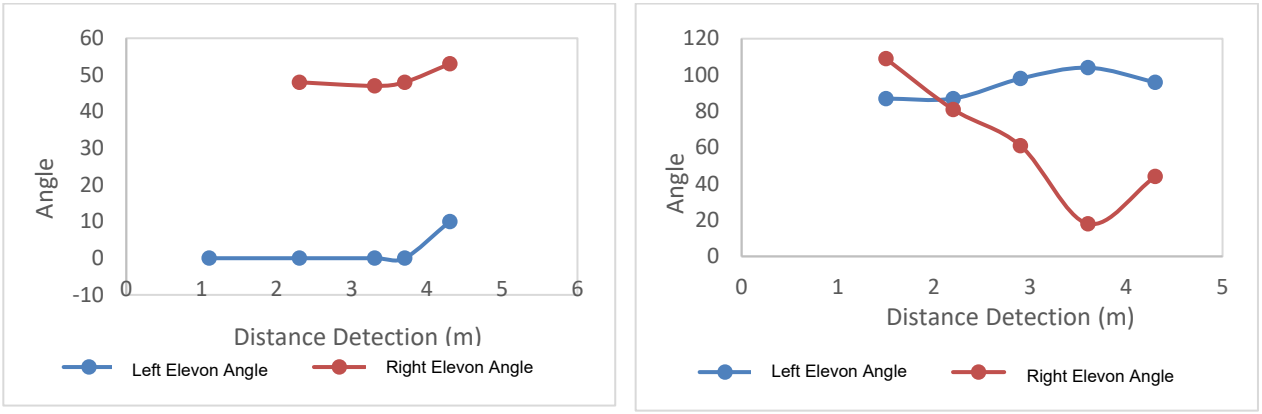


Figure 6. Comparison of average FPS for each camera type using YOLOv8 and YOLOv11.

3.4 Elevon Angle Response

The elevon angle response was analyzed to observe actuator outputs generated from the detection results and the corresponding control values. Based on the results presented in Figure 7, K4 with YOLOv11 showed a left elevon angle that tended to remain at 0° for most of the tested distances, while the right elevon angle ranged from 41° to 53°. This indicates that the actuator response for this combination was predominantly expressed through the right elevon. For K5 with YOLOv11, the left elevon angle ranged from 87° to 104°, while the right elevon angle varied from 109° to 18°. The larger variation in the right elevon angle indicates a more pronounced actuator correction in response to changes in the detected target position. These results indicate that variations in camera type may affect not only detection and distance estimation performance but also the control outputs transmitted to the elevon actuators.

Figure 7 presents the elevon angle responses for two camera–model combinations selected as representative examples, namely K4 and K5 with YOLOv11. The selection of these two cameras does not imply that K1, K2, and K3 were excluded from the overall control-system evaluation. All camera types were evaluated during the detection experiments, and the corresponding control parameters were recorded during testing. However, to maintain a concise graphical presentation and facilitate interpretation of the actuator response patterns, the elevon-angle visualization was focused on K4 and K5. These two cameras were selected because they exhibited different performance characteristics: K4 produced lower distance-estimation errors, whereas K5 provided a higher processing speed when using YOLOv11. Therefore, Figure 7 is presented as a representative illustration of how variations in detection results can propagate to differences in the resulting elevon responses.



(a) (b)

Figure 7. Left and right elevon angle responses to detection results for (A) Logitech BCC950–YOLOv11 and (B) Logitech HD720P–YOLOv11.

3.5 Camera Technical Factors Affecting Camera Performance

Differences in camera performance are not solely associated with video resolution but may also be influenced by technical characteristics such as sensor capability, optical quality, image processing, and frame-rate stability. These differences can affect the quality of the images received by the YOLO model and subsequently influence detection and distance estimation results. In this study, camera K4 with its specified resolution produced the lowest MAE and RMSE when combined with YOLOv11, whereas K5 produced a higher FPS. These findings indicate that higher video resolution does not directly guarantee higher processing speed, as system performance is also affected by device characteristics and the image acquisition process.

3.6 Comparison with Previous Studies and Discussion

The results obtained from the YOLOv8 and YOLOv11 experiments indicate that UAV detection system performance is influenced not only by the detection model but also by the characteristics of the camera device used as the image source. This finding is consistent with the studies by Yilmaz and Kutbay [6] and Huang et al. [7], which reported that YOLO development and optimization can improve UAV detection capabilities. Studies by Fang et al. [8] and Li et al. [9] also demonstrated that improvements to YOLO architectures can enhance UAV object detection, particularly for small objects. However, these studies primarily focused on developing or optimizing detection models, whereas variations in image acquisition devices were not the primary focus. In the present study, both YOLO models were evaluated using five camera devices with different characteristics under the same testing conditions. The results demonstrated differences in distance estimation performance and processing speed among the cameras. For example, with YOLOv11, K4 produced an MAE of 0.34 m and an RMSE of 0.39 m but achieved only 4.8 FPS. In contrast, K5 produced an MAE of 0.38 m and an RMSE of 0.43 m with an FPS of 15.8. These results indicate that higher processing speed does not necessarily correspond to lower distance estimation error. These findings extend the scope of previous research by indicating that UAV detection system evaluation should consider the image acquisition device in addition to the detection algorithm. Unlike studies that primarily compare or develop YOLO architectures, this study maintains the same detection models and testing parameters while comparing multiple camera devices as image sources. This approach allows the observed differences to be used to examine the influence of camera characteristics on distance estimation, FPS, and UAV control outputs. From an application perspective, the results indicate that camera selection should be considered according to the requirements of the system. Cameras with higher FPS can support faster image processing, whereas cameras with smaller distance estimation errors can provide more accurate distance information. In this study, K5 with YOLOv11 produced an

FPS of 15.8 with an MAE of 0.38 m and an RMSE of 0.43 m, whereas K4 with YOLOv11 produced an MAE of 0.34 m and an RMSE of 0.39 m with an FPS of 4.8. Therefore, camera selection in a real-time UAV vision system should consider the trade-off between processing speed and distance estimation accuracy.

4 Conclusion

Based on the experimental results, variations in camera type affected drone detection performance and UAV control response in systems based on YOLOv8 and YOLOv11. Differences in camera characteristics resulted in variations in distance estimation error, processing speed, and control outputs, including roll, pitch, and elevon angles. Overall, the K4–YOLOv11 combination produced the lowest distance estimation error, with an MAE of 0.34 m and an RMSE of 0.39 m, but it also exhibited the lowest processing speed at 4.8 FPS. In contrast, the K5–YOLOv11 combination produced a relatively low distance estimation error, with an MAE of 0.38 m and an RMSE of 0.43 m, while achieving a substantially higher processing speed of 15.8 FPS. These results demonstrate that distance estimation performance and processing speed do not necessarily improve simultaneously.

The experimental results also indicate that variations in detection results affect the roll and pitch control values and the left and right elevon angles generated by the system. Therefore, camera characteristics should be considered as an important component in vision-based drone detection and tracking systems. For real-time drone detection applications that require a balance between distance estimation performance and processing speed, the results obtained in this study indicate that K5 provides a more balanced performance, whereas K4 provides smaller distance estimation errors but with lower processing speed.

Acknowledgments

The authors would like to express their appreciation to the Air Force Research and Development Agency (*Dinas Penelitian dan Pengembangan TNI Angkatan Udara—Dislitbangau*) for providing research support and facilities, particularly the camera devices used for image acquisition and drone detection system testing. This support greatly contributed to the implementation of the research and the preparation of the results presented in this article.

Referensi

[1] J. Jessin, C. Heinzlef, N. Long, and D. Serre, “A systematic review of UAVs for island coastal environment and risk monitoring: Towards a resilience assessment,” *Drones*, vol. 7, no. 3, 2023. https://doi.org/10.3390/drones7030206
[2] A. Alsumayt et al., “Detection of malicious UAVs based on secure authentication over the Internet of Drones (IoD),” *Multimedia Tools and Applications*, vol. 85, no. 3, 2026. https://doi.org/10.1007/s11042-026-21468-3
[3] P. Jiang, X. Yang, Y. Wan, T. Zeng, M. Nie, and Z. Liu, “DRBD-YOLOv8: A lightweight and efficient anti-UAV detection model,” *Sensors*, vol. 24, no. 22, 2024. https://doi.org/10.3390/s24227148
[4] A. V. R. Katkuri, H. Madan, N. Khatri, A. S. H. Abdul-Qawy, and K. S. Patnaik, “Autonomous UAV navigation using deep learning-based computer vision frameworks: A systematic literature review,” *Array*, vol. 23, 2024. https://doi.org/10.1016/j.array.2024.100361
[5] B. Prayoga, B. Paradis Anwar, I. Arrizqi, T. Munanzar Aliek, and P. Rosyani, “Deteksi Kendaraan dengan Metode YOLO,” *Jurnal AI dan SPK: Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, vol. 3, no. 2, pp. 177–185, 2025. [Online] Available : https://jurnalmahasiswa.com/article/3274
[6] B. Yilmaz and U. Kutbay, “YOLOv8-based drone detection: Performance analysis and optimization,” *Computers*, vol. 13, no. 9, 2024. https://doi.org/10.3390/computers13090234
[7] M. Huang, W. Mi, and Y. Wang, “EDGS-YOLOv8: An improved YOLOv8 lightweight UAV detection model,” *Drones*, vol. 8, no. 7, 2024. https://doi.org/10.3390/drones807033
[8] A. Fang, S. Feng, B. Liang, and J. Jiang, “Real-time detection of unauthorized unmanned aerial vehicles using SEB-YOLOv8s,” *Sensors*, vol. 24, no. 12, 2024. https://doi.org/10.3390/s24123915
[9] J. Li, J. Zhang, Y. Shao, and F. Liu, “SRE-YOLOv8: An improved UAV object detection model utilizing Swin Transformer and RE-FPN,” *Sensors*, vol. 24, no. 12, 2024. https://doi.org/10.3390/s24123918
[10] J. Su, Y. Qin, Z. Jia, and B. Liang, “MPE-YOLO: Enhanced small target detection in aerial imaging,” *Scientific Reports*, vol. 14, no. 1, 2024. https://doi.org/10.1038/s41598-024-68934-2
[11] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, “A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Machines*, vol. 11, no. 5, 2023. https://doi.org/10.3390/make5040083

[12]Y. Zheng, Y. Jing, J. Zhao, and G. Cui, "LAM-YOLO: Drones-based small object detection on lighting-occlusion attention mechanism YOLO," 2024., *arXiv:2411.00485*, [Online]. Available: http://arxiv.org/abs/2411.00485

[13]Z. Du, X. Feng, F. Li, Q. Xian, and Z. Jia, "A lightweight UAV visual obstacle avoidance algorithm based on improved YOLOv8," Computers, Materials & Continua, vol. 81, no. 2, pp. 2607–2627, 2024. https://doi.org/10.32604/cmc.2024.056616

[14] A. Barisic, M. Car, and S. Bogdan, "Vision-based system for a real-time detection and following of UAV," in Proc. 2019 Int. Workshop on Research, Education and Development on Unmanned Aerial Systems (RED-UAS), pp. 156–159, 2019. https://doi.org/10.1109/REDUAS47371.2019.8999675

[15]A. Ayuningtyas, I. Riadi, and A. Yudhana, "The effect of light intensity, camera pixel quality, camera distance, and object altitude on detection accuracy in a real-time drone surveillance system using YOLOv5," *Journal of Innovation Information Technology and Application (JINITA)*, vol. 7, no. 2, pp. 234–246, 2025. https://doi.org/10.35970/jinita.v7i2.2843