Implementation of Deep Learning in Automatic Enemy Object Shooting to Assist in Securing Military Guard Posts

Putfui Ari Dewi[1], Hadiwiyatno Hadiwiyatno[2*], Waluyo Waluyo[3], Amalia Eka Rakhmania[4]

[1,3] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

[2]Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

[1]putfuiari18@gmail.com, [2]hadiwiyatno@polinema.ac.id, [3]waluyo@polinema.ac.id, [4]amaliaeka.rakhmania@polinema.ac.id

Abstract— **Military guard posts play a crucial role in securing strategic areas, particularly in high-risk regions that frequently face armed threats. However, limited infrastructure and equipment often reduce security effectiveness. This study aims to develop an automated security system using deep learning technology for enemy detection and automatic shooting responses. The system employs Faster R-CNN algorithm for object recognition and classification, integrated with a Logitech C920 camera for image capture, Raspberry Pi 4 Model B for data processing, and Arduino Nano to control servo motors for weapon movement and firing. Testing results demonstrate that the system effectively distinguishes between friend and enemy objects based on clothing attributes under various lighting conditions. The servo motor achieves 87.25 percent accuracy for horizontal movement, 88.89 percent for vertical movement, and 88.87 percent shooting accuracy with an average response time of 1.011 seconds. Detection probability reaches 85.11 percent under bright conditions and decreases to 65.24 percent in dark environments. The system successfully detects enemies up to 36 meters distance. This automated system enhances guard post security and reduces risks for military personnel.**

Keywords— Automatic Shooting, Deep Learning, Faster R-CNN, Guard Post Security System, Weapon Guidance.

I. INTRODUCTION

The military strength of a country plays a strategic role in maintaining sovereignty and national security. This is a key concern for Indonesia, which consistently strives to enhance its military capabilities. This progress is evident in Indonesia's military ranking, which is currently 13th out of 145 countries [1], along with a 19% increase in the defense budget compared to 2021 [2]. The Indonesian military, known as Tentara Nasional Indonesia (TNI), serves as the nation's defense and security force, with an organizational structure equivalent to a ministry. It consists of the Indonesian Army (TNI-AD), Navy (TNI-AL), and Air Force (TNI-AU). The Indonesian Army (TNI-AD) is responsible for ground defense operations, including the crucial task of guarding military outposts [3].

Military guard posts are key elements in securing strategic areas, monitoring, and controlling access to sensitive locations such as military bases and border areas. However, these posts are often targeted by armed groups or unidentified individuals, particularly in Indonesia's border regions. Several major attacks were reported in Papua in 2023, including assaults by the Armed Criminal Group (KKB) in Kenyam, Gome, Aroba, and South Aifat districts, which resulted in casualties, including two TNI soldiers, one of whom was Corporal Herdianto, who lost his life [4]-[7].

Guard posts in Papua face challenges related to infrastructure and security. Most posts are located in remote areas with limited facilities [8]. To enhance security, TNI has begun utilizing modern technologies, such as drones for patrol and surveillance. However, the application of Artificial Intelligence (AI) in guard post security has yet to be fully optimized.

Deep Learning-based technology offers significant potential to improve guard post security by enabling automatic object recognition and detection. Deep Learning, a branch of Machine Learning, utilizes neural networks to analyze complex data patterns [9]. This technology allows the identification of friend or enemy objects with high accuracy, thereby minimizing risks for security personnel.

Previous research has demonstrated the effectiveness of AI technology in military systems. For example, Haar Cascade and SSD Inception-V2 have been used for real-time target recognition and tracking, while the PID algorithm has been applied for precise control in automated systems [10], [11]. Pattern recognition and IoT-based sentry gun systems can replace listening posts [12], while the Smart Shooter System has achieved 99% accuracy in tracking targets up to 15 meters [13]. The PID algorithm enables shooting accuracy with a minimal error rate of 0.6% [10], [14].

*Corresponding author

Additionally, the HSV method has been used to detect red-colored objects with an 83.3% success rate at distances up to 290 cm, though performance decreases under low-light conditions [15]. Image processing allows the recognition of TNI camouflage uniforms and Protap helmets at distances of 2–12 meters with a detection angle of 0°–60° [16]. SSD Inception-V2 has also been applied to detect targets and prevent accidental shooting with a response time of 0.27 seconds, although it is limited to specific weapon types [11]. Moreover, wireless control using stepper motors and NRF24L01 antennas provides flexibility in shooting training [17]. Based on these methods, this study proposes the development of a Deep Learning-based military guard post security system using the Faster R-CNN algorithm for object recognition. The system is integrated with servo control for target tracking and automatic shooting, where camera perspectives are converted into servo movement angles.

II. METHODS

A. Research Flowchart

The type of research conducted is aimed at the development of science and technology. The flowchart shown in Fig. 1 systematically explains the stages involved in the research or system development process.

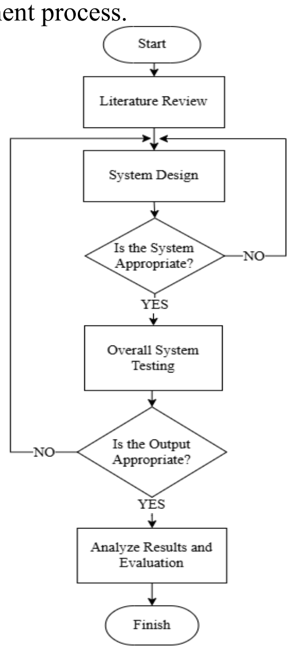


Figure 1. Research flowchart

The system development process, as shown in Figure 1, begins with a literature review to gather information and references related to concepts, methods, and technologies relevant to system development. Based on this information, the system design phase is carried out, which includes defining hardware, software, and the overall workflow diagram. Once the design is established, system evaluation is conducted to verify its alignment with initial requirements and specifications. If discrepancies are found, necessary revisions are made to improve system accuracy. Following evaluation, system integration testing ensures the proper functionality of both hardware and software components, confirming that they operate seamlessly together. The next phase involves output analysis and evaluation, where testing results are examined to determine whether the system output meets predefined specifications. If deviations are detected, further adjustments are implemented. Finally, performance analysis and evaluation measure the system's effectiveness based on parameters such as accuracy, efficiency, and overall functionality, ensuring that the system operates optimally under different conditions.

B. Block Diagram

The workflow shown in Fig. 2 begins with capturing object images, followed by data processing to detect and classify objects as "friend" and "enemy", and concludes with weapon movement control.

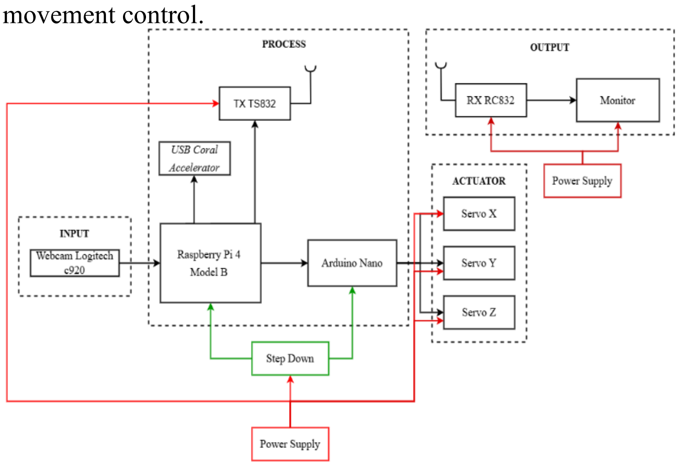


Figure 2. System block diagram

The system consists of several key components categorized into input, processing, actuator, and output stages, all working together to enable autonomous detection and response. In the input stage, a Logitech C920 webcam captures real-time images for object recognition, providing a continuous video feed for analysis. The processing stage utilizes a Raspberry Pi 4 Model B, which applies deep learning algorithms, specifically Faster R-CNN, to detect and classify enemy objects based on predefined characteristics. Once an enemy is identified, the Raspberry Pi transmits control data to an Arduino Nano, which functions as a microcontroller to regulate the movement of three servo motors (X, Y, and Z) for precise targeting. Additionally, a TS832 video sender transmits processed video data for real-time monitoring, ensuring situational awareness.

The actuator stage comprises three servo motors: Servo Motor X controls the horizontal movement of the weapon, Servo Motor Y adjusts the vertical positioning, and Servo Motor Z is responsible for triggering the firing mechanism with precise timing. Finally, in the output stage, an RS832 video receiver collects the transmitted video data, which is then displayed on a monitor. This enables real-time visualization of detected objects, ensuring accurate weapon tracking and system responsiveness while allowing operators to monitor and assess the system's performance in dynamic environments.

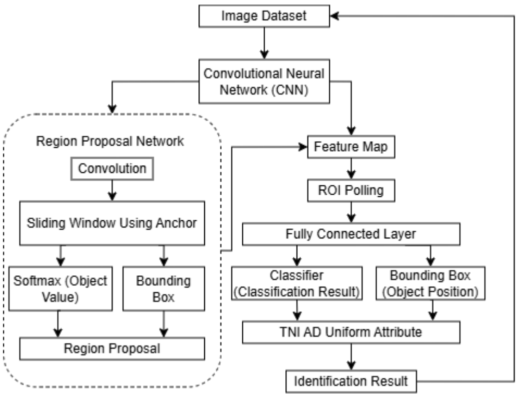


Figure 3. Object identification system block diagram

The system, as shown in Fig. 3, starts with a dataset consisting of 3000 images categorized into "friend" (TNI AD uniform) and "enemy" (casual clothing). This dataset is used to train and test the Faster R-CNN model. The workflow involves feature extraction using Convolutional Neural Networks (CNNs), producing feature maps that highlight important visual representations of the images.

The training process employs a batch size of 20 across 20 epochs, resulting in 3,000 iterations to optimize the model's parameters. During training, the Region Proposal Network (RPN) identifies potential object regions by extracting features through convolutional layers, applying a sliding window approach with anchors to detect candidate object areas, and using a softmax function to compute the probability of object presence. Regions that meet the predefined threshold are highlighted with bounding boxes to indicate object locations. Next, the Region of Interest (ROI) Pooling layer extracts relevant portions from the feature maps, classifying objects as either "friend" or "enemy" based on probability scores and determining their positions using bounding boxes. At the final stage, detected objects exhibiting camouflage patterns are classified as "friend," while those wearing casual clothing are identified as "enemy."

The system outputs classification labels, bounding boxes, and confidence levels. Overall, the model achieves high classification accuracy, optimized through 20 epochs of training to ensure precise differentiation between "friend" and "enemy."

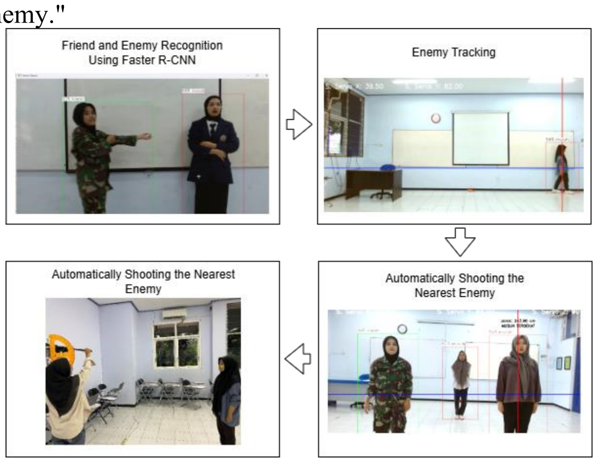


Figure 4. Real-time system block diagram

The system shown in Fig. 4 uses the Faster R-CNN algorithm to detect and recognize objects as "friend" or "enemy" based on images captured by the camera. Detected "enemy" objects are tracked using the bounding box geometry by calculating the center point of the bounding box to maintain the target's position and identity. The system then calculates the distance of each "enemy" object from the camera to prioritize the nearest target. Once the nearest target is identified, the horizontal and vertical servo motors automatically align the weapon with the target, and the trigger servo activates the firing mechanism. This process is executed in real-time to ensure the system can detect, track, and neutralize the nearest enemy effectively.

C. System Schematic Diagram

The system consists of multiple interconnected hardware components designed to facilitate real-time object detection, classification, and response. The schematic diagram, as shown in Fig. 5, provides a detailed representation of the system's hardware architecture and its interactions.

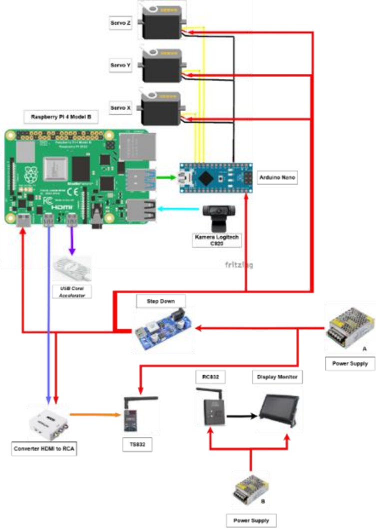


Figure 5. System schematic diagram

The Raspberry Pi 4 Model B serves as the primary controller, At the core of the system, the Raspberry Pi 4 Model B serves as the primary processing unit, executing deep learning-based object detection and decision-making tasks. The Arduino Nano functions as a secondary controller, responsible for operating three servo motors that adjust the weapon's position and trigger mechanism. The system employs a Logitech C920 camera for real-time image and video acquisition, while a TS832 video transmitter sends processed video data from the Raspberry Pi to a remote display. The RC832 video receiver captures this data and displays it on a monitor via an HDMI-to-RCA converter, which converts the signal format for compatibility.

To enhance performance, a USB Coral Accelerator is integrated, significantly improving the efficiency of deep learning computations for object recognition. The Arduino Nano controls the servo motors as follows: Servo X for horizontal weapon positioning, Servo Y for vertical alignment, and Servo Z for trigger activation.

Power for the Raspberry Pi, Arduino Nano, and servo motors are supplied by an external power source. A step-down converter module ensures the voltage is regulated to meet the system's specific requirements. The Raspberry Pi 4 Model B, acting as a mini-PC, processes image recognition tasks using deep learning models, relays control commands to the Arduino Nano via serial communication, and coordinates weapon alignment and firing mechanisms with precision.

D. Device Flowchart

The device flowchart, as shown in Fig. 6, visually represents the sequential process of the system's operation. It provides a clear depiction of how the system initializes, detects objects, classifies them as "friend" or "enemy," and executes the appropriate response based on the detected object.

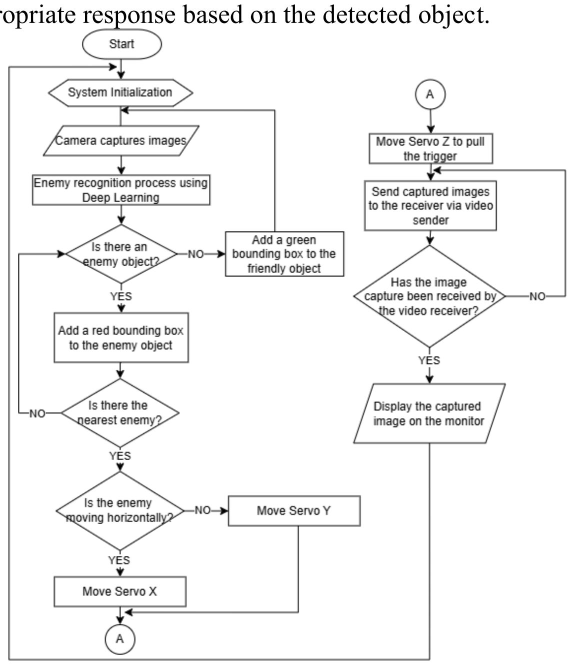


Figure 6. Device flowchart

The system initialization process begins by preparing all components for operation. The camera is first activated to ensure readiness for capturing images, followed by verifying the microcontroller's status for data processing. Once initialized, the system proceeds to object recognition, where the camera captures an image and processes it using a Deep Learning algorithm based on Faster R-CNN. This algorithm analyzes the image to detect and classify objects as either "friend" or "enemy."

If no enemy object is detected, the system marks friendly objects with a green bounding box and loops back to the image capture step. However, if an enemy is identified, the system highlights it with a red bounding box and determines its position relative to the camera. The nearest enemy object is then classified, and the system evaluates whether it is moving horizontally. If the classification process fails, the system reattempts enemy detection.

For enemy objects with no significant horizontal movement, Servo Y adjusts the vertical aim of the weapon. Conversely, if the enemy moves horizontally, Servo X adjusts accordingly. Once the enemy's position is accurately determined and both servos have aligned the weapon, Servo Z activates to pull the trigger.

After firing, the captured image is transmitted to the receiver via a video transmitter. The system then verifies whether the transmitted image has been successfully received. If transmission fails, the system resends the image until successful. Once received, the image is displayed on the monitor, allowing for real-time visual monitoring. This process repeats continuously, ensuring a seamless workflow of image capture, object recognition, and result transmission.

III. RESULTS AND DISCUSSION

A. Prototype Development Results

Fig. 7 presents the hardware prototype developed using a toy weapon replica. This prototype is designed to move both horizontally and vertically using servo motors as actuators.

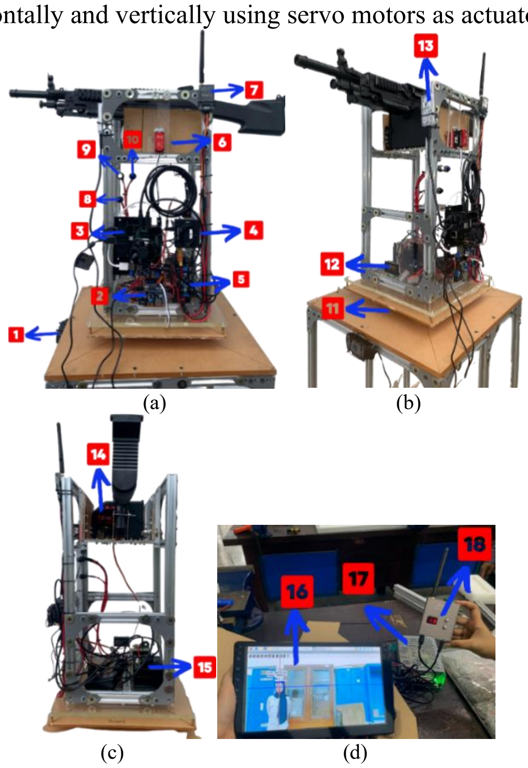


Figure 7. (a) Side view of the prototype components; (b) Front view of the prototype components; (c) Rear view of the prototype components; (d) Display on the monitor

The system consists of several components, including:

1. Logitech Camera: Captures images as input for object detection.
2. Arduino Nano: Functions as the actuator controller for the device.
3. Raspberry Pi 4 Model B: Serves as a mini-PC for processing captured images.
4. HDMI to RCA Converter: Converts the video signal from the Raspberry Pi to the TS832 module.
5. Step-Down Converter: Reduces voltage from 12V to 5V for system operation.
6. Servo Y: Controls the vertical movement of the weapon.

7. TS832: Wireless video transmission module for real-time image streaming.
8. TS832 Power Button (Black): Powers the TS832 module.
9. System Power Button (White): Activates the overall system.
10. Servo Control Button (Blue): Activates servo motor movement.
11. Servo X: Controls the horizontal movement of the weapon.
12. LCD Display: Shows the power status of the device.
13. USB Coral Accelerator: Enhances AI-based data processing for object detection.
14. Servo Z: Activates the weapon's trigger mechanism.
15. Motorcycle Battery (12V, 3.5Ah): Provides the main power supply for the system.
16. Monitor Screen: Displays the detection results and processed data.
17. Power Supply: Powers the RC832 module and the monitor screen.
18. RC832: Wireless video receiver module for real-time image reception.

B. System Testing

The system testing focuses on the implementation of Deep Learning to detect and classify objects as either "friend" or "enemy," determine the nearest enemy as the primary threat, and control the weapon system to perform automatic targeting and firing. As illustrated in Figure 8, the detection results are displayed on the monitoring screen, where bounding boxes and labels indicate the classification outcomes. In this stage, the system continuously tracks detected objects in real time and calculates their relative distances from the weapon system. Based on this calculation, the nearest enemy is selected as the priority target. Fig. 8(a) shows the monitoring display where the detected enemy is highlighted and tracked, while servo motors on the X and Y axes are adjusted accordingly to align the weapon. Subsequently, as shown in Fig. 8(b), the weapon physically moves toward the identified target, demonstrating the integration between the deep learning detection module and the mechanical actuation system.

(a) (b)

Figure 8. (a) Monitor screen display; (b) Weapon movement

The testing scenarios summarized in Table 1 evaluate the system's performance under various conditions, including different object positions, movements, and quantities. When an object is classified as a "friend," the system maintains the current servo position, ensuring that no targeting or firing action is taken. Conversely, when an "enemy" is detected, the system evaluates its spatial position relative to other detected objects and dynamically adjusts the servo motors to aim at the closest enemy before executing automatic firing. The scenarios include static targets, moving targets, parallel object arrangements, and sequential configurations involving multiple enemies. In situations with more than one enemy present, the system consistently prioritizes the nearest threat, engaging it first before shifting focus to subsequent targets. This behavior demonstrates that the system is capable of accurate classification, reliable target prioritization, and responsive weapon control in diverse operational scenarios.

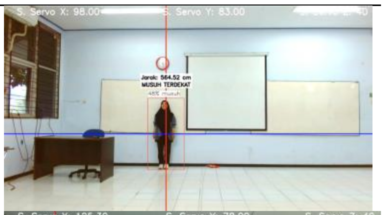
TABLE I
SYSTEM TESTING IN DETECTION

No	Testing Scenario	Position	Testing Description
1.		Standing	The detected object is a "friend." The servo does not move because the system does not aim the weapon at the "friend" object.
2.		Walking	The detected object is a "friend." The servo does not move because the system does not aim the weapon at the "friend" object.
3.		Parallel	The detected objects are a "friend" and a "foe" in a standing parallel position. The servo moves the weapon to aim at the "foe" object and fires automatically.
4.		Sequential	The detected objects are a "foe" in the front position and a "friend" in the back position. The servo moves the weapon to aim at the nearest "foe" and fires automatically.
5.		Sequential	The detected objects are one "friend" and one "foe" in parallel positions, with another "foe" behind them. The servo moves the weapon to aim at the nearest "foe" in the front position and fires automatically.
6.		Sequential	The detected objects are three "foes" in sequential positions. The servo moves the weapon to aim at the nearest "foe" and fires automatically.

C. Conditions During Testing

The lighting conditions during the testing are shown in Table 2. The tests were conducted both indoors and outdoors under different lighting conditions: indoor (bright), indoor (dim), indoor (dark), and outdoor (bright). In dark conditions, the environment was not entirely pitch black, as there was ambient light from surrounding sources or outdoor lamps. These tests were conducted to assess the impact of lighting conditions on the system's detection accuracy. The results from these tests show how well the system performs under different lighting conditions and whether additional adjustments, such as infrared sensors or brightness compensation, are required to improve detection accuracy in low-light environments.

TABLE II
TESTING RESULTS UNDER DIFFERENT LIGHTING CONDITIONS

Lighting Condition	Measurement Results	Monitor Display
Indoor (Bright)		
Indoor (Dim)		
Indoor (Dark)		
Outdoor (Bright)		

D. *Testing of Servo Targeting Movement and Shooting*

The servo targeting movement and shooting tests were conducted to evaluate the system's performance in automatically aiming the weapon at the detected target position captured by the camera.

TABLE III
TESTING OF WEAPON TARGETING AND SHOOTING

Distance (m)	Condition		Movement 1	2	3	4	5	6	7	*Error* (%)	Accuracy (%)
2	X	Weapon(°)	160	115	110	90	65	55	20	19.16	80.84
		Target (°)	130	120	110	90	80	70	60		
	Y	Senjata(°)	80	80	80	80	80	80	80	11.11	88.89
		Weapon(°)	90	90	90	90	90	90	90		
	Z		neck	neck	chin	forehead	cheek	cheek	chin	0	100
4	X	Weapon(°)	160	126	116	90	70	50	20	13.38	86.62
		Target (°)	140	125	110	90	70	55	40		
	Y	Weapon(°)	80	80	80	80	80	80	80	11.11	88.89
		Target (°)	90	90	90	90	90	90	90		
	Z		-	Stomach	Stomach	Stomach	Waist	Stomach	Stomach	14.29	85.71
6	X	Weapon(°)	160	150	120	90	80	45	20	12.64	87.36
		Target (°)	150	135	105	90	65	45	30		
	Y	Weapon(°)	80	80	80	80	80	80	80	11.11	88.89
		Target (°)	90	90	90	90	90	90	90		
	Z		-	-	Chest	Leg	Thigh	Stomach	Chest	28.57	71.43
Overall Average										13.48	86.51

Table 3 shows the measurement of servo movement angles along the X-axis (horizontal), Y-axis (vertical), and Z-axis (shooting), as well as the target positions at different test distances. The tests were conducted at distances of 2 m, 4 m, and 6 m, with test subjects wearing casual clothing. The collected data includes the alignment of the weapon angle with the target position, error rates in servo movement, and the system's accuracy in detecting enemy body positions for executing shots.

E. *Enemy Detection and Shooting Test*

Table 4 presents the system's capability to detect enemies and perform automatic shooting under various indoor lighting conditions, including bright (139 lux), dim (45 lux), and dark (0 lux). The test was conducted at distances of 2 m, 4 m, and 6 m, with test subjects wearing casual clothing to evaluate the system's reliability in detecting enemies and executing automated shots.

TABLE IV
ENEMY DETECTION AND SHOOTING TEST IN INDOOR ENVIRONMENTS

Lighting Condition	Distance (m)	Indicator	Movement 1	2	3	4	5	6	7	*Error* (%)	Accuracy (%)
Bright (139 lux)	2	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	4	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	6	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	-	-	28.57	71.43
Dim (45 lux)	2	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	4	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	-	14.29	85.71
	6	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	-	-	28.57	71.43
Dark (0 lux)	2	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	4	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	-	14.29	85.71
	6	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	-	-	28.57	71.43
Overall Average										6.35	93.65

TABLE V
ENEMY DETECTION AND SHOOTING TEST IN OUTDOOR ENVIRONMENTS

Lighting Condition	Distance (m)	Indicator	Movement 1	2	3	4	5	6	7	*Error* (%)	Accuracy (%)
Bright (4013 lux)	2	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	4	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	✓	✓	0	100
	6	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	✓	✓	✓	✓	✓	-	-	28.57	71.43
	8	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	-	✓	✓	✓	✓	-	-	42.86	57.14
	10	Enemy	✓	✓	✓	✓	✓	✓	✓	0	100
		Shoot	-	-	✓	✓	-	-	-	71.43	28.57
Overall Average										14.29	85.71

Table 5 shows the results of enemy detection and shooting tests conducted outdoors under bright lighting conditions of 4013 lux. The test was performed at distances of 2 m, 4 m, 6 m, 8 m, and 10 m to evaluate the system's detection capability limits in high-intensity lighting and open environments. Test subjects also wore casual clothing. The results provide insights into the system's performance across varying distances and lighting conditions compared to indoor testing.

F. *Inference Time Testing Before and After Using TPU*

The inference time test was conducted to compare performance before and after using the USB Coral Accelerator as shown in Table 6. Initially, the model was tested without the TPU, using the default CPU or GPU of the Raspberry Pi 4 Model B, the model configuration was set to `model='PROJECT2024_3000.tflite'` and `edgetpu='0'`. Subsequently, the model was tested with the USB Coral Accelerator, using a TPU-compatible TensorFlow Lite model `model='PROJECT2024_3000_edgetpu.tflite'` and `edgetpu='1'`.

TABLE VI
INFERENCE TIME BEFORE AND AFTER USING TPU

Trial	Inference Time (ms)	
	Before Using TPU	After Using TPU
1	685.14	119.62
2	703.79	124.20
3	683.65	168.59
4	681.62	280.65
5	682.96	313.73
6	681.15	284.27
7	707.82	273.25
8	681.3	116.03
9	668.47	129.15
10	663.86	130.69
11	664.32	107.64
12	664.89	101.15
13	660.74	230.64
14	660	257.58
15	682.47	222.81
16	663.39	278.39
17	663.18	287.94
18	663.33	260.96
19	662.57	268.89
20	665.72	217.96
Average	674.51	208.71

The results indicate that using the USB Coral Accelerator significantly enhances inference speed, achieving an average reduction of 69.05% in inference time.

G. *Maximum Shooting Distance and Shooting Response Time Testing*

The maximum shooting distance test was conducted by firing gel ball bullets at targets at various distances. The results are shown in Table 7, where the term "Hits Target" indicates that the bullet successfully reached the target directly, while ″Bounced" indicates that the gel bullet no longer had enough energy to reach the target and merely bounced upon contact. Meanwhile, Table 8 presents the results of the shooting response time test. This test was conducted 10 times to determine how quickly the shooting system could fire at the target.

TABLE VII
MAXIMUM SHOOTING DISTANCE TESTING

Distance (m)	Accuracy
2	Hits Target
4	Hits Target
6	Hits Target
8	Hits Target
10	Hits Target
12	Bounced
14	Bounced

TABLE VIII
SHOOTING RESPONSE TIME TESTING

Trial	Response Time (s)
1	1.09
2	1.10
3	0.87
4	0.95
5	0.83
6	0.89
7	0.98
8	1.27
9	1.05
10	1.08
Average	1.011

H. *Effect of Light Intensity and Distance on Detection Accuracy*

The analysis of the probability values based on light intensity and distance provides insight into how environmental conditions affect the system's enemy detection performance. As shown in Table 9, lighting conditions have a significant influence on detection accuracy. Under bright lighting conditions (139 lux), the system achieves the highest detection probability of 85.11%, indicating that sufficient illumination allows the deep learning model to extract visual features more effectively, such as object shape, contours, and movement. When the lighting intensity decreases to dim conditions (45 lux), the detection probability drops to 78.04%, showing a moderate decline in performance due to reduced visual clarity. In completely dark conditions (0 lux), the detection probability further decreases to 65.24%, which highlights the system's limitation when visual information is minimal or absent. This reduction occurs because the camera-based detection relies heavily on visible light to distinguish objects, making low-light environments more challenging for accurate classification.

Table 10 presents the effect of distance on enemy detection probability under constant bright lighting conditions. At a distance of 2 m, the system records the highest probability value of 85.12%, indicating optimal detection when the target is relatively close to the camera. As the distance increases to 4 m, the detection probability decreases to 78.23%, and further drops to 71.63% at 6 m. This trend demonstrates that increasing distance reduces the resolution and size of the detected object within the image frame, making feature extraction and classification more difficult. Overall, the results from both tables indicate that optimal system performance is achieved under good lighting conditions and shorter distances, while reduced illumination and greater distances significantly affect detection reliability.

TABLE IX
ENEMY DETECTION PROBABILITY BASED ON LIGHTING CONDITIONS

Lighting Condition	Probability Value (%)
Bright (139 lux)	85.11
Dim (45 lux)	78.04
Dark (0 lux)	65.24

TABLE X
ENEMY DETECTION PROBABILITY BASED ON DISTANCE

Distance (m)	Probability Value (%)
2	85.12
4	78.23
6	71.63

IV. CONCLUSION

This study demonstrates that the Deep Learning-based system successfully detects and classifies "friend" and "enemy" objects based on clothing attributes under various lighting conditions and complex scenarios. The system can automatically move servo X (87.25%) and servo Y (88.89%) to aim the weapon at the nearest enemy and perform automatic shooting with 88.87% and a response speed of 1.011 seconds. Distance and lighting factors influence system performance. Enemy detection is optimal up to 36 m, while friend detection reaches 16 m. Detection accuracy decreases with changes in lighting conditions: 85.11% at 139 lux (bright), 78.04% at 45 lux (dim), and 65.24% at 0 lux (dark). Accuracy also declines with increased distance: 85.12% at 2 m, 79.32% at 4 m, and 71.63% at 6 m. Automatic shooting achieves 100% under bright (139 lux) and dim (45 lux) lighting at 2–4 m, but drops to 71.43% at 6 m. In dark conditions (0 lux), the average shooting accuracy is 71.43%. Outdoor performance decreases at longer distances, with 90.46% at 2, 4, and 6 m, dropping to 57.14% at 8 m and 28.57% at 10 m. These findings confirm that integrating Deep Learning and servo control effectively enhances military guard post security, although further optimization is required for extreme conditions.

REFERENCES

[1] Natasha Ancely, "Kekuatan Militer Indonesia Berada di Peringkat ke 13 dari 145 Negara." Accessed: Jun. 02, 2024. [Online]. Available: https://www.kompas.tv/video/474913/kekuatan-militer-indonesia-berada-di-peringkat-ke-13-dari-145-negara

[2] A. Ahdiat, "Belanja Militer Indonesia Meningkat, tapi Rasionya Tak Sampai 1% PDB," Databoks.Katadata.Co.Id.

[3] R. Wardi, "2016, TNI AD Segera Bentuk Dua Kodam Baru," BERITASATU.COM. Accessed: May 04, 2024. [Online]. Available: https://www.beritasatu.com/news/336091/2016-tni-ad-segera-bentuk-dua-kodam-baru

[4] N. Katingka, "KKB Serang Pos TNI di Papua Barat Daya, Satu Prajurit Gugur," Harian Kompas. Accessed: Apr. 02, 2024. [Online]. Available: https://www.kompas.id/baca/nusantara/2023/12/26/kkb-serang-markas-tni-di-papua-barat-daya-satu-prajurit-gugur

[5] F. M. L. Costa, "Takut Terdampak Serangan KKB, Ratusan Warga Nduga Mengungsi," Harian Kompas. Accessed: Apr. 02, 2024. [Online]. Available: https://www.kompas.id/baca/nusantara/2023/06/01/takut-serangan-kkb-ratusan-warga-nduga-diungsikan-ke-ibu-kota-nduga

[6] Rahman, "Mencekam! Pos TNI di Distrik Aroba Papua Barat Diberondong Tembakan OTK," SINDONEWS.COM. Accessed: Apr. 02, 2024. [Online]. Available: https://daerah.sindonews.com/read/1208119/174/mencekam-pos-tni-di-distrik-aroba-papua-barat-diberondong-tembakan-otk-1695398979

[7] R. Latumahina, "KKB Serang Pos TNI di Puncak Papua Tengah: 1 Prajurit dan 3 Teroris Luka," detikNews. Accessed: Apr. 02, 2024. [Online]. Available: https://news.detik.com/berita/d-6879075/kkb-serang-pos-tni-di-puncak-papua-tengah-1-prajurit-dan-3-teroris-luka

[8] D. M. Nirmala Maulana Achmad, "TNI AD Akan Perbaiki Pos-pos Prajurit di Papua , KSAD Maruli : Pakai Terpal , Mau Jualan Bakso ? Ironi Denis Murib , Pilih Apa Itu Desertir TNI , Papua Adalah Eks Praju … Pengkhianatan Denis." Accessed: Jun. 18, 2024. [Online]. Available: https://nasional.kompas.com/read/2024/02/29/18331341/tni-ad-akan-perbaiki-pos-pos-prajurit-di-papua-ksad-maruli-pakai-terpal-mau

[9] S. M. Rezkia, "Apa Itu Machine Learning? Kenali Perbedaan Machine Learning dan *Deep Learning*," pp. 1–10, 2020, [Online]. Available: https://www.dqlab.id/pahami-machine-learning-untuk-hadapi-industri-data-sebenarnya

[10] S. Sabila, P. dani Prasetyo adi, S. Subairi, and M. Baidlowi, "Sistem Otomatis Penggerak Lesan Tembak Untuk Latihan Teknik Tempur Cepat Dengan Metode PID," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 3, no. 2, pp. 63–70, 2021, doi: 10.26905/jasiek.v3i2.8373.

[11] E. N. Ufuk Asıl, "Using Image Processing Techniques to Increase Safety in Shooting Ranges," *Int. J. Comput. Sci. Enginering an Appl.*, vol. 11, pp. 1–6, 2021, doi: 10.5121/ijcsea.2021.11103.

[12] C. Simatupang, M. Anwar, R. Budi, P. Angkatan, and D. Pushubad, "Rancang Bangun Sistem Tracking Pada Sentry Gun Sebagai Pengganti Pos Dengar Gun As a Replacement of Hearing Posts Based on," vol. 3, pp. 84–92, 2022.

[13] A. Zani, "Smart Shooter System pada Robot Tank dengan Realtime Target Recognition and Tracking Berbasis Raspberry Pi," *J. Telkommil*, pp. 1–8, 2020, [Online]. Available: https://journal.poltekad.ac.id/index.php/kom/article/view/45

[14] A. Raharjo, E. Kuncoro, and I. Azhar, "Rancang Bangun Tracking Arah Tembakan Menggunakan Sensor Posisi Berbasis Pid," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 7, no. 1, pp. 43–48, 2021, doi: 10.15575/telka.v7n1.43-48.

[15] S. Muharom and B. Prasetyo, "Automatics Detect and Shooter Robot Based on Object Detection Using

Camera," *Prz. Elektrotechniczny*, vol. 1, no. 1, pp. 50–54, 2021, doi: 10.15199/48.2022.01.07.

[16] B. Harijanto, A. Muzakki, M. Syafaat, and A. Militer, "Implementasi Pengenalan Musuh Melalui Loreng TNI dan Helm Protap Berbasis Image Processing Pada Robot Penjaga Bersenjata Otomatis SS2," *J. Elkasista*, vol. 5, 2023.

[17] C. Grigoraş, S.-M. Petrişor, C.-T. Codreanu, and E.-M. Borboană, "Examples of Good Educational Practices on the Modernization of Shooting Sessions through the Experimental Construction of a Functional Model of a Shooting Range and of a Mobile Target Device Operated by Radio Waves," *Sci. Bull.*, vol. 27, no. 2, pp. 112–122, 2022, doi: 10.2478/bsaft-2022-0013.