**Journal of Applied Informatics and Computing (JAIC)**
Vol.10, No.4, August 2026, pp. 3523~3534
e-ISSN: 2548-6861 

# Real-Time Weapon Detection and Suspect Face Capturing System Using YOLOv8

**Chairina Ulfa [1]*, Muhammad Fikry [2]**, Hafizh Al Kautsar Aidilof [3]***
* Teknik Informatika, Universitas Malikussaleh
chairina.220170043@mhs.unimal.ac.id [1], muh.fikry@unimal.ac.id [2], hafizh.alkautsar@unimal.ac.id [3]

**Article Info**

***Article history:***

Received 2026-06-26
Revised 2026-07-25
Accepted 2026-08-08

***Keyword:***

*Computer Vision,*
*Firearms,*
*Object Detection,*
*Sharp Weapons,*
*YOLOv8.*

**ABSTRACT**

The rise of violent crimes involving sharp weapons and firearms in public spaces, including educational campuses, demands an automated real-time surveillance system to assist security personnel. This study proposes a web-based weapon detection system using YOLOv8, specifically designed to detect seven object classes: sickle, machete, axe, sword, knife, pistol, and rifle. When a weapon is detected, the system automatically captures the suspect's facial image using Haar Cascade and triggers alarm notifications, detection logs, and statistical reports. This integrated data package serves as critical digital evidence to support post-incident identification and investigation. To train the model, we constructed a dataset of 11,445 images sourced from public datasets, video frame extraction, and smartphone camera captures, which was subsequently augmented to 27,687 images to enhance model generalization. The evaluation results demonstrate strong performance with a Precision of 94.3%, Recall of 87.8%, mAP@0.5 of 93.2%, and mAP@0.5:0.95 of 60.1%. Real-time testing at distances ranging from 50 cm to 500 cm confirmed that the system reliably detects most weapon classes, particularly achieving consistent detection for sickles, machetes, and rifles across all tested ranges, while performance for smaller objects like knives and pistols showed decreased accuracy at extreme distances, indicating directions for future work. The findings confirm that the proposed system effectively detects and classifies sharp weapons and firearms in real-time while simultaneously providing visual documentation of the perpetrator, offering a practical and comprehensive security solution for campus environments.



## I. Introduction

Violent crime remains a major public safety issue worldwide, with incidents involving sharp weapons and firearms continuing to increase in both frequency and severity. The presence of dangerous weapons significantly elevates the risk of fatal injuries and poses serious threats to public security. According to the Indonesian National Police Criminal Information Center, approximately 325,150 criminal cases were recorded in Indonesia in 2024. This number increased substantially to approximately 414,812 reported cases during the period of January 1 to December 24, 2025, indicating a persistent rise in criminal activities and highlighting the need for more effective preventive security measures [1], [2].

Indonesia has established strict regulations governing the possession and use of firearms and other dangerous weapons through Emergency Law No. 12 of 1951. Unauthorized possession or use of such weapons is subject to severe criminal penalties due to the significant threat they pose to public safety [3]. Nevertheless, illegal weapon-related incidents continue to occur in public places, including educational institutions, demonstrating that legal regulations alone are insufficient without effective preventive monitoring systems.

University campuses are expected to provide a safe and conducive learning environment for students, lecturers, and staff. However, campuses are not immune to criminal activities such as personal conflicts, bullying, gender-based violence, and physical assaults. Reports indicate that more than 14,000 cases of violence involving university students were documented in Indonesia during 2024 [4]. Furthermore, the open accessibility and high mobility within campus environments make security monitoring increasingly challenging. Although many universities have implemented surveillance systems using Closed-Circuit Television (CCTV), conventional CCTV primarily functions as a passive recording device and depends heavily on continuous human observation. Consequently, potential threats may not be detected promptly, reducing the effectiveness of early intervention [5].

The urgency of developing intelligent surveillance systems has become more evident following several violent incidents occurring on university campuses. One notable case occurred in early 2026 when a student attacked another student using an axe during a thesis defense session, resulting in serious injuries and psychological trauma among the academic community [6]. Such incidents illustrate that dangerous situations may arise unexpectedly, even in environments generally regarded as secure. These events also reveal the limitations of conventional surveillance systems, which are incapable of automatically recognizing dangerous objects before an attack occurs. Therefore, proactive surveillance systems capable of providing real-time early warnings are becoming increasingly important.

Recent advances in Artificial Intelligence (AI), particularly Deep Learning, have significantly improved the capability of computer vision systems for automatic object detection. Among the available object detection algorithms, You Only Look Once (YOLO) has become one of the most widely adopted approaches because it performs object localization and classification simultaneously within a single neural network, enabling high detection accuracy while maintaining real-time processing speed [7]. The latest version, YOLOv8, further enhances detection performance through architectural improvements that provide better accuracy, faster inference, and greater flexibility across various computer vision applications.

Previous studies have demonstrated the effectiveness of deep learning techniques for object detection tasks. Herlangga developed a traditional weapon classification model using EfficientNetB3 with transfer learning and achieved nearly perfect classification performance across training, validation, and testing datasets [8]. Similarly, Setyawan and Widiyanto implemented YOLOv8 for safety shoe detection, obtaining a Precision of 97%, Recall of 94.9%, and a mean Average Precision (mAP) of 97.5%, while maintaining an average processing time of approximately 0.175 seconds [9]. Supriyanto also reported high performance using YOLOv7 for safety helmet detection, achieving an Accuracy of 97.23%, Precision of 98.71%, and Recall of 95.63%, demonstrating that modern YOLO architectures can operate efficiently even on resource-constrained hardware platforms [10].

Although these studies confirm the effectiveness of YOLO-based object detection, most existing research has focused on detecting a single object category, such as safety equipment or a specific weapon type. Consequently, limited attention has been given to developing an integrated surveillance system capable of simultaneously detecting multiple categories of dangerous weapons under real-world conditions. Moreover, many previous studies emphasize detection performance without incorporating practical security features that support immediate response after a dangerous object has been identified.

To address these limitations, this study proposes a real-time dangerous weapon detection system based on YOLOv8 using transfer learning. Unlike previous approaches, the proposed system is capable of detecting seven weapon categories simultaneously, including knife, machete, sword, sickle, axe, pistol, and rifle. The developed web-based system not only performs real-time object detection but also integrates automatic face capture, confidence threshold verification, alarm activation, image evidence storage, and detection logging as an early warning mechanism. These integrated functionalities are intended to improve situational awareness and assist security personnel in responding more rapidly to potential threats within campus environments.

The main contributions of this research are threefold. First, an integrated YOLOv8-based surveillance system capable of detecting multiple categories of dangerous weapons in real time is developed. Second, the proposed system combines intelligent object detection with automatic evidence collection and alert generation to support proactive security monitoring. Third, the performance of the proposed model is comprehensively evaluated using standard object detection metrics, including Precision, Recall, F1-score, mAP, and inference speed, to demonstrate its feasibility for real-time campus surveillance applications. Unlike previous studies that mainly focused on object detection accuracy, this work emphasizes the integration of multi-class weapon detection with automatic evidence collection and real-time security response within a single surveillance framework.

## II. METHOD

### A. Research Framework

The research workflow involves data collection and preparation (annotation, preprocessing, augmentation, and dataset splitting), YOLOv8 model training and evaluation, and final system integration with real-time testing.

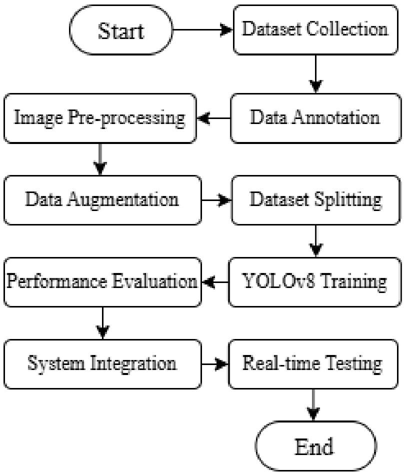


Figure 1. Research Framework

## *B. Dataset Collection*

Parameter ValueThe dataset used in this study was acquired from multiple sources to improve data diversity and enhance the robustness of the proposed weapon detection model under real-world conditions. The dataset consists of publicly available images collected from Kaggle and Roboflow Universe, manually selected high-resolution images from Pinterest, video frames extracted from YouTube, and images captured directly using an Android smartphone camera. The direct image acquisition was performed at distances ranging from 1 to 5 meters under various indoor and outdoor backgrounds to simulate realistic surveillance scenarios.

Combining data from multiple sources enables the model to learn different object appearances, viewpoints, lighting conditions, backgrounds, and scales, thereby improving its generalization capability during real-time detection. The seven weapon classes, namely sickle, machete, axe, sword, knife, pistol, and rifle, were selected because they represent both sharp weapons and firearms that are frequently associated with violent incidents and public security threats. These weapon categories may potentially appear in surveillance environments such as educational institutions, public facilities, and other crowded areas. Therefore, the selected classes were considered sufficiently representative of the threat scenarios targeted in this study.

In total, 10,649 images were collected. During the annotation process, a total of 11,445 weapon instances were labeled because several images contained more than one weapon object. The distribution of annotated weapon instances for each class is presented in Table 1. Before model training, all images were processed using the Roboflow platform for manual annotation, image preprocessing, data augmentation, and dataset partitioning.

TABEL 1
DATASET COMPOSITION

| Weapon Class | Number of Annotated Objects |
|---|---|
| Sickle | 1,643 |
| Machete | 1,653 |
| Axe | 1,678 |
| Sword | 1,677 |
| Knife | 1,649 |
| Pistol | 1,494 |
| Rifle | 1,651 |
| Total | 11,445 |

## *C. Data Annotation*

After the dataset acquisition stage, all collected images were manually annotated using the Roboflow platform. Each weapon object was labeled by drawing a bounding box around its visible region and assigning the corresponding class label. The annotation process covered seven weapon categories: sickle, machete, axe, sword, knife, pistol, and rifle.

To ensure annotation consistency, each image was carefully reviewed so that every visible weapon was accurately labeled with a single bounding box. The annotated dataset was then exported in the YOLO format, which stores the object class along with the normalized coordinates of the bounding box center, width, and height. This annotation format is directly compatible with the YOLOv8 framework and serves as the ground truth during the model training process [11].

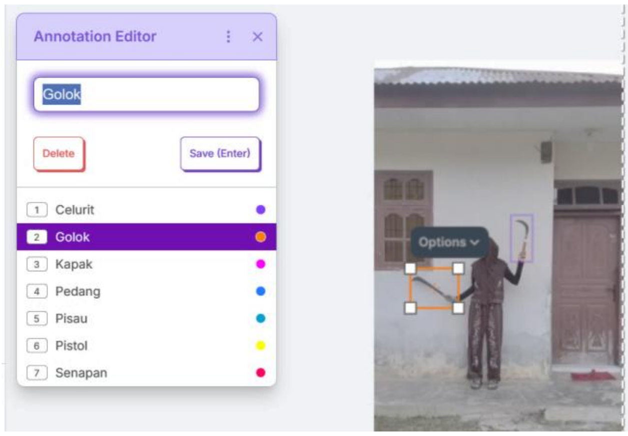


Figure 2. Example of weapon image annotation using Roboflow.

## *D. Image Preprocessing*

Image preprocessing was performed to improve image quality and standardize the input data before training the YOLOv8 model. Three preprocessing techniques were applied using the Roboflow platform: Auto-Orient, Resize, and Auto-Adjust Contrast.

The Auto-Orient operation automatically corrected the image orientation based on the embedded metadata, ensuring that all images were aligned consistently regardless of the device used during image acquisition. Subsequently, all images were resized to 640 × 640 pixels, which corresponds to the default input resolution of the YOLOv8 model. Standardizing the image size reduces computational complexity while maintaining sufficient visual information for object detection.

To further improve image quality, Auto-Adjust Contrast based on the Adaptive Histogram Equalization (AHE) method was applied. This technique enhances local image contrast by redistributing pixel intensity values, making weapon objects more distinguishable under varying illumination conditions. Consequently, the preprocessing stage improves feature visibility and contributes to more robust object detection performance during model training.

### *E. Data Augmentation*

Several augmentation techniques were employed, including horizontal flipping, random rotation within a range of −15° to +15°, brightness adjustment between −15% and +15%, grayscale conversion, and random cropping. Horizontal flipping enables the model to recognize weapon objects from different orientations, while random rotation improves robustness against variations in camera viewpoints. Brightness adjustment simulates different illumination conditions commonly encountered in real-world surveillance environments. Grayscale augmentation was applied to 15% of the training images by converting RGB images into grayscale, encouraging the model to focus on structural and shape-related features rather than color information. In addition, random cropping was performed by removing up to 20% of the original image area, enabling the model to learn from partially visible objects and improving its robustness against partial occlusion.


Figure 3. Example of image augmentation using random rotation

### *F. Data Splitting*

After preprocessing, the 10,649 images were partitioned into training, validation, and testing subsets using an 80:10:10 ratio. The dataset consisted of 8,519 training images, 1,065 validation images, and 1,065 testing images.

Data augmentation was applied exclusively to the training subset to increase data diversity, while the validation and testing subsets remained unchanged. This strategy prevents data leakage by ensuring that augmented versions of training images were never included in the validation or testing datasets.

After augmentation, the number of training images increased from 8,519 to 25,557, whereas the validation and testing subsets remained unchanged at 1,065 images each. Consequently, the final dataset consisted of 27,687 images, including 25,557 training images (92%), 1,065 validation images (4%), and 1,065 testing images (4%), as summarized in Table 2.

TABEL 2
DATASET DISTRIBUTION AFTER AUGMENTATION

| Dataset | Images | Percentage |
|---|---|---|
| Training | 25,557 | 92% |
| Validation | 1,065 | 4% |
| Testing | 1,065 | 4% |
| Total | 27,687 | 100% |

### *G. Training Configuration*

The proposed model was initialized using the pretrained YOLOv8n weights provided by the Ultralytics framework. The model was trained on the annotated dataset specified in the data.yaml configuration file for a total of 100 epochs using transfer learning. The training process was conducted incrementally by resuming from the latest checkpoint (last.pt) until the predefined number of epochs was completed. The best-performing model (best.pt) obtained during training was subsequently used for the implementation and evaluation of the proposed real-time weapon detection system.

The dataset configuration stored in the data.yaml file specifies the directory paths for the training, validation, and testing subsets, as well as the seven weapon classes used in this study. Table 3 summarizes the training configuration employed for the YOLOv8 model.

TABEL 3
TRAINING CONFIGURATION

| **Parameter** | **Value** |
|---|---|
| Model | YOLOv8n (Pretrained) |
| Dataset Configuration | data.yaml |
| Epochs | 100 |
| Batch Size | 16 |
| Image Size | 640 × 640 |
| Optimizer | Auto |
| Initial Learning Rate | 0.01 |
| Momentum | 0.937 |
| Weight Decay | 0.0005 |
| Warm-up Epochs | 3 |
| Patience | 50 |
| Mixed Precision | Enabled (AMP) |
| Output Model | best.pt |

### *H. Face Capture Integration*

After the YOLOv8 model detects one or more weapon objects in a video frame, the system selects the weapon with the highest confidence score to trigger the warning mechanism, including the alarm, screenshot capture, and logging process. Subsequently, the same frame is processed using the Haar Cascade classifier to detect frontal human faces. The detected faces are automatically cropped by adding a 20% margin around the face region before being stored together with the corresponding detection log and screenshot. This integration enables the system to record facial evidence immediately after a weapon is detected without requiring additional image acquisition.

When multiple faces appear in a single frame, the Haar Cascade detector identifies all visible frontal faces, and each detected face is cropped and saved individually. Likewise, when multiple weapons are detected simultaneously, the system uses the weapon with the highest confidence score to activate the alarm, while the screenshot still preserves all detected weapon objects within the frame. However, the current implementation does not associate a particular face with a specific detected weapon. Instead, all detected faces are stored as supporting evidence for further inspection.

Haar Cascade was selected because it provides fast detection with relatively low computational requirements and can be easily integrated into the proposed real-time weapon detection system. Compared with more recent face detection methods such as RetinaFace, MTCNN, and YOLO-Face, Haar Cascade generally offers lower detection accuracy under challenging conditions, including extreme illumination, large head pose variations, and partial occlusion. Nevertheless, since face detection in this study is intended only to capture facial evidence after weapon detection rather than perform face recognition, Haar Cascade provides an appropriate trade-off between computational efficiency and detection performance, making it suitable for real-time implementation.

Future work may replace Haar Cascade with more advanced face detectors, such as RetinaFace or YOLO-Face, to improve robustness under challenging surveillance conditions while maintaining the overall integration framework proposed in this study.

### I. *Yolov8-Based Weapon Detection*

The proposed weapon detection model was developed using the YOLOv8 algorithm due to its high accuracy and real-time detection capability. This section describes the object detection process and the YOLOv8 architecture employed in this study.

1) *Object Detection*: Object detection is a fundamental computer vision task that involves both identifying instances of specific objects within an image or video and precisely locating them. Unlike image classification, which merely assigns a label to an entire picture, object detection pinpoints the position of each object by drawing bounding boxes around them and assigning a corresponding class label, such as "person," "vehicle," or "animal." Modern object detection heavily relies on deep learning architectures, particularly Convolutional Neural Networks (CNNs). These methods are generally categorized into two-stage detectors (like the R-CNN family), which prioritize high accuracy, and single-stage detectors (like YOLO and SSD), which prioritize real-time speed and efficiency. This technology is the backbone of numerous real-world applications, including autonomous driving systems, facial recognition, medical imaging analysis, retail inventory tracking, and security surveillance [12], [13].

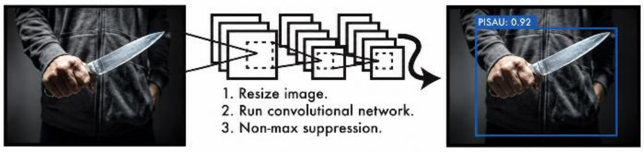


Figure 4. YOLOv8 object detection workflow.

As illustrated in Fig. 4, the detection process begins by resizing the input image to the predefined input size before being processed through the convolutional neural network. The model then predicts the bounding boxes, object classes, and confidence scores. Finally, Non-Maximum Suppression (NMS) is applied to remove overlapping predictions, resulting in the final detection output.

2) YOLOv8 Architecture: The YOLOv8 architecture, released by Ultralytics in early 2023, represents a significant evolution in real-time object detection. It is structured around three core components: the Backbone, Neck, and Head

The Backbone is responsible for extracting hierarchical feature representations from the input image. Compared with previous YOLO versions, YOLOv8 adopts the C2f (Cross-Stage Partial bottleneck with two convolutions) module, replacing the earlier C3 module to enhance feature extraction and improve gradient propagation while maintaining computational efficiency. In addition, a Spatial Pyramid Pooling Fast (SPPF) module is employed to capture multi-scale contextual information and enlarge the receptive field [14].

The Neck integrates feature maps from different network levels using a combination of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). This feature fusion mechanism combines high-level semantic information with low-level spatial details, allowing the model to detect objects of various sizes more effectively [12].

The Head performs the final prediction by estimating the object class, bounding box coordinates, and confidence score. Unlike previous anchor-based YOLO models, YOLOv8 adopts an anchor-free detection head, which directly predicts object locations without predefined anchor boxes. Furthermore, the decoupled head separates the classification and localization tasks, resulting in faster convergence and improved detection accuracy across multiple object scales [15], [16].

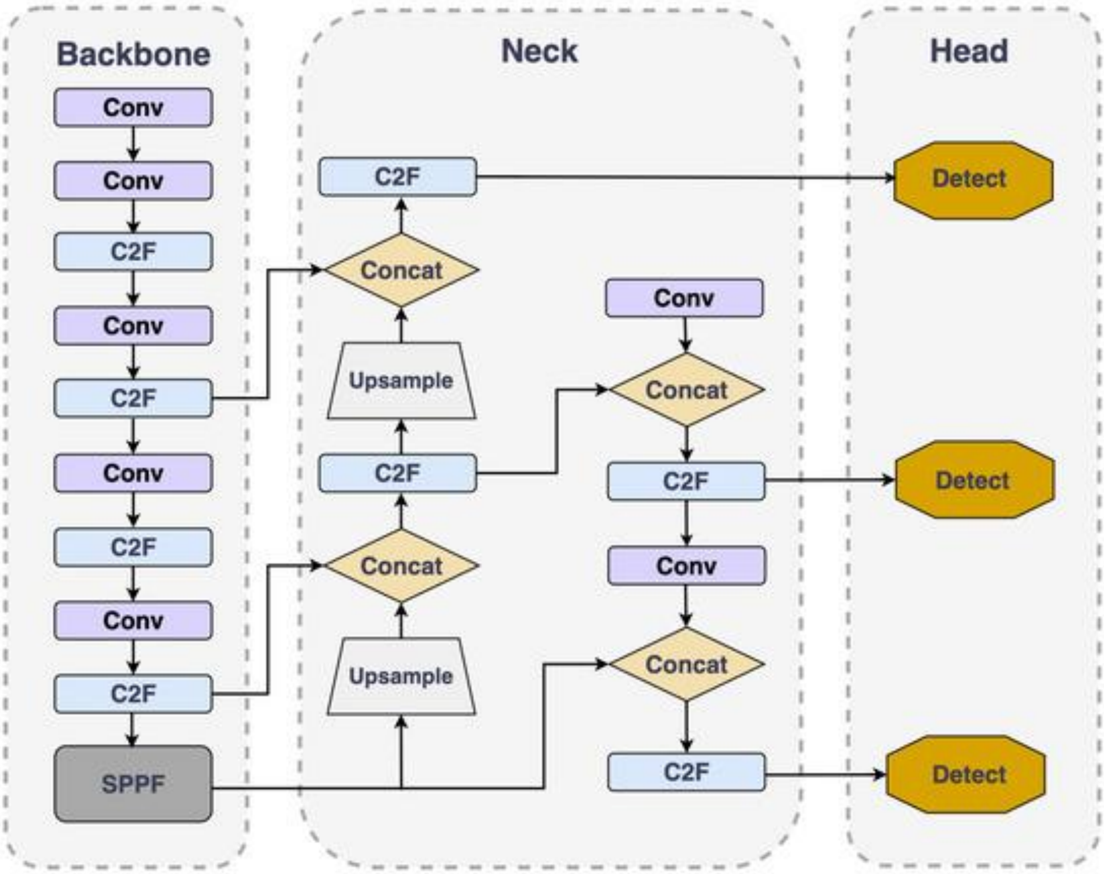


Figure 5. Architecture of the YOLOv8 model.

*J. Comparison with Previous Studies*

TABEL 4
COMPARISON WITH PREVIOUS WEAPON DETECTION STUDIES

| Study | Model | Detection Classes | Accuracy (%) | Recall (%) | Precision (%) | F1-score (%) |
|---|---|---|---|---|---|---|
| Achmad et al. [17] | YOLOv6 | Weapon, | 90% | 100% | 83% | 91% |
| Achmad et al. [17] | Faster R-CNN | Weapon, | 80% | 87% | 76% | 81% |
| Achmad et al. [17] | SSD | Weapon, | 83% | 87% | 76% | 84% |
| This Study | YOLOv8 | Seven weapon classes | 82.52% | 90.23% | 91.90% | 90.54% |

Table 4 shows a comparison between the proposed method and the previous study conducted by Achmad et al. Although the proposed model achieved a slightly lower overall accuracy than the YOLOv6 model reported by Achmad et al., this comparison should be interpreted carefully because the proposed system addresses a substantially more challenging seven-class weapon detection task while simultaneously integrating automatic face capture, alarm activation, detection logging, and web-based monitoring. Detecting multiple weapon categories with varying shapes, sizes, and visual characteristics is inherently more challenging than detecting a single weapon class. Despite this increased complexity, the proposed model achieved competitive performance, with a recall of 90.23%, precision of 91.90%, and F1-score of 90.54%. These results demonstrate that the proposed system provides a more comprehensive and practical solution for real-time surveillance applications.

## III. Results and Discussion

### A. Model Training Results

1) *Precision-Recall (PR) Curve:* The performance of the YOLOv8 model was further evaluated in depth by analyzing four main metric curves, namely F1-Confidence, Precision-Confidence, Precision-Recall, and Recall-Confidence.

As shown in Figure 6, the Precision-Recall curve achieves an overall mAP of 0.96 across all seven classes at a confidence threshold of 0.000, confirming that the model maintains an excellent balance between false positives and missed detections. Furthermore, the Precision-Confidence and F1-Confidence curves both reach their peak values of 1.00 at a confidence threshold of 0.912. This convergence indicates that the optimal inference confidence for deployment should be configured around 0.91 to simultaneously maximize both precision and recall, minimizing false alarms while ensuring most true objects are captured.

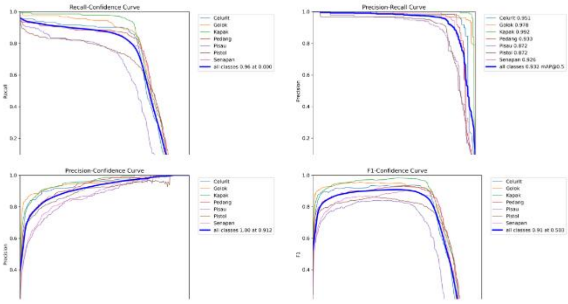


Figure 6. Performance curves of the YOLOv8 model

maintains an excellent balance between false positives and missed detections. Furthermore, the Precision-Confidence and F1-Confidence curves both reach their peak values of 1.00 at a confidence threshold of 0.912. This convergence indicates that the optimal inference confidence for deployment should be configured around 0.91 to simultaneously maximize both precision and recall, minimizing false alarms while ensuring most true objects are captured

2) *Losses and Map:* Evaluation of the YOLOv8 model training process was conducted by observing the performance metric trends over 100 epochs. The analysis of model performance development during both the training and validation phases was comprehensively reviewed based on changes in loss, Precision, Recall, and Mean Average Precision (mAP) values.

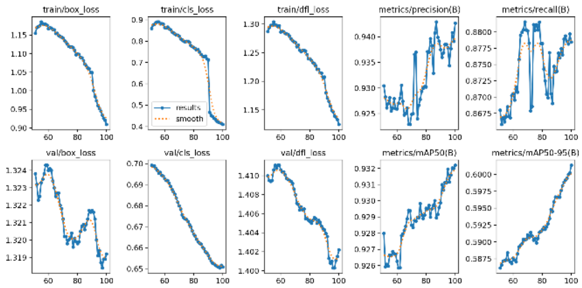


Figure 7. Training loss and mAP curves of the YOLOv8 model

The loss curves in Figure 7 demonstrate consistent and stable convergence throughout the training process, with the validation loss closely tracking the training loss. This behavior indicates that the model generalized effectively to unseen data without significant overfitting. Concurrently, the mAP curves show a steep initial improvement within the first 50 epochs, gradually plateauing toward the end of training. The final mAP values reaching 95.1% (Celurit) and 99.2% (Kapak) at IoU=0.5 confirm that the YOLOv8 backbone successfully extracted discriminative features. However, the noticeably lower mAP@0.5:0.95 values (averaging 30–35% lower) across all classes reveal that accurate bounding box regression is more challenging, especially for objects with extreme aspect ratios or small scales.

TABEL 5
PER-CLASS MAP RESULTS

| Class | mAP@0.5 (%) | mAP@0.5:0.95 (%) |
|---|---|---|
| Axe | 99.2 | 69.1 |
| Machete | 97.8 | 68.1 |
| Sickle | 95.1 | 57.7 |
| Sword | 93.3 | 64.6 |
| Rifle | 92.6 | 58.8 |
| Knife | 87.2 | 45.9 |
| Pistol | 87.2 | 56.8 |

3) *Confusion Matrix*: The confusion matrix is a tabular visualization that compares the predicted class labels against the ground-truth labels for each test sample. It provides a detailed diagnostic tool to identify specific inter-class misclassifications (e.g., which classes are frequently confused with one another) and the rate of false positives against background regions. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate errors [18].

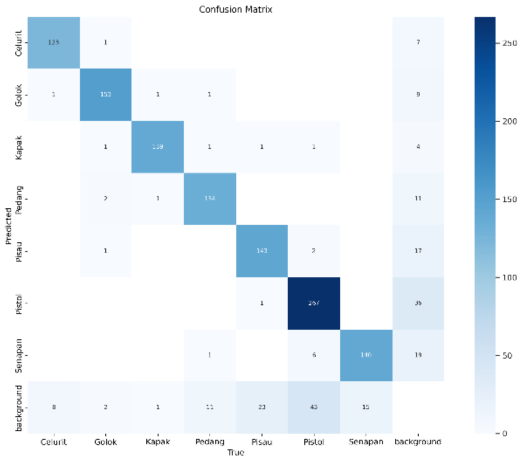


Figure 8. Confusion Matrix

The confusion matrix in Figure 8 reveals several critical error patterns. The Senapan class exhibits substantial confusion with Pistol, with 23 instances of rifles misclassified as handguns, likely due to their shared dark metallic textures and similar silhouettes when viewed from a distance. The Pisau class is occasionally misclassified as Pedang (2 instances) and vice versa (1 instance), reflecting their inherent visual similarity as bladed objects with elongated shapes. The most alarming observation is the extreme false-positive rate against background regions for the Celurit class, where the model incorrectly labels 250 background patches as Celurit. This indicates that curved, sickle-like contours frequently appear in natural outdoor environments (e.g., vegetation, farming tools, or curved architectural elements), causing excessive spurious detections. Conversely, the Kapak class demonstrates minimal cross-class confusion, with only 1 misclassification to Golok and 1 to Pedang, confirming its superior discriminability due to its rigid and uniquely distinguishable geometric structure. The diagonal values (e.g., 267 for Pistol, 151 for Golok, 139 for Kapak) show the absolute number of correct predictions per class, which align with the high recall values that will be quantified in the next subsection.

TABEL 6
ACCURACY, PRECISION, RECALL, F1-SCORE PER-CLASS

| Class | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Axe | 92.67 | 97.89 | 94.56 | 96.20 |
| Machete | 88.95 | 95.63 | 92.73 | 94.16 |
| Sickle | 87.86 | 93.18 | 93.89 | 93.53 |
| Sword | 82.72 | 90.54 | 90.54 | 90.54 |
| Rifle | 77.35 | 90.32 | 84.34 | 87.23 |
| Pistol | 75.00 | 83.70 | 87.83 | 85.72 |
| Knife | 76.06 | 85.12 | 87.73 | 86.41 |

As shown in Table 6, the *Axe* class achieved the best detection performance, with an Accuracy of 92.67%, Precision of 97.89%, Recall of 94.56%, and an F1-Score of 96.20%. In contrast, the *Pistol* and *Knife* classes exhibited relatively lower Accuracy values, indicating a higher rate of misclassification or missed detections. Overall, the proposed model demonstrated strong detection and classification performance across all weapon classes on the test dataset.

4) *Analysis of Bounding Box Prediction Results:* To further evaluate the qualitative performance of the proposed model, a visual analysis of the detection results was conducted on the testing dataset. The analysis focuses on the model's ability to accurately localize and classify different weapon categories under various image conditions, including variations in object size, viewpoint, background, and illumination. Figure 10 illustrates representative examples of the bounding box prediction results produced by the YOLOv8 model.

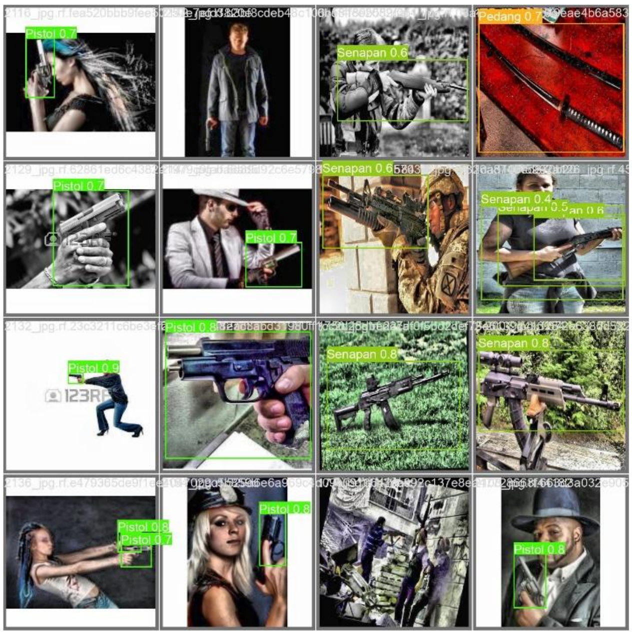

Figure 9. Examples of bounding box prediction results on the testing dataset.

The model successfully detected various weapon classes, including pistol, rifle, and sword, with confidence scores ranging from 0.30 to 0.90. Most predictions generated bounding boxes that closely matched the object boundaries, indicating accurate object localization. Higher confidence scores were generally achieved when the weapon appeared clearly in the image with sufficient size and minimal occlusion.

Several detections exhibited relatively lower confidence scores, particularly when the weapon was partially occluded, located farther from the camera, or captured under challenging lighting conditions. Nevertheless, the model was still able to correctly classify most weapon objects despite these variations. Overall, the qualitative results demonstrate that the proposed YOLOv8 model provides reliable localization and classification performance, supporting its application for real-time weapon detection.

### B. *Real-Time System Implementation*

The trained YOLOv8 model was integrated into a web-based surveillance system to perform real-time weapon detection. The system was developed using the Flask framework as the backend, which processes video streams from either a webcam or uploaded video files and performs frame-by-frame inference using the trained YOLOv8 model. Detection results, including bounding boxes, class labels, and confidence scores, are transmitted to the web interface for real-time visualization.

The web-based dashboard provides several features to support security monitoring, including live video streaming, input source selection, detection controls, camera status monitoring, detection history, statistical summaries, automatic face capture, and an audible alarm. As illustrated in Fig. 10, the dashboard enables operators to monitor detection results while recording every detection event for further analysis.

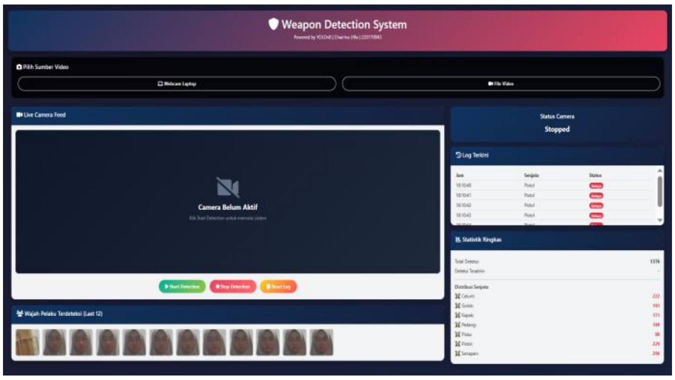

Figure 10. Web-based surveillance dashboard with all features

When a weapon is detected with a confidence score above the predefined threshold, the system automatically displays the corresponding bounding box and class label, activates an alarm after continuous detection for more than three seconds, and captures the face of the detected individual. The captured image, together with the detection information, is subsequently stored in the database and displayed on the dashboard. Experimental implementation demonstrated that the system operated at approximately 25–30 FPS on a standard laptop equipped with an Intel Core i3-8130U processor without a dedicated GPU, indicating that the proposed system is suitable for real-time deployment on low-cost hardware. Although CPU utilization and memory consumption were not quantitatively measured, the successful execution on a low-end Intel Core i3 processor without a dedicated GPU indicates that the proposed system requires relatively modest computational resources for real-time deployment.

### C. *Field Testing Results*

Real-time field testing was conducted in a campus corridor at distances ranging from 50 cm to 500 cm. The purpose of this test was to determine the maximum effective range for each weapon class and to identify distance thresholds beyond which detection becomes unreliable. Such information is critical for practical camera placement. Each of the seven weapon classes was tested at 10 distance points. Detection was categorized as:

✓ = Detected (consistent bounding box for at least 1 second)
~ = Unstable detection (bounding box appears intermittently)
✗ = Not detected


*Figure 11. Sample field test detections at different distances*

Based on the sample detections, the system successfully recognized weapons at close range (50–150 cm) with high confidence scores (typically 0.7–0.9). As distance increased, the bounding boxes became smaller and confidence scores gradually decreased, especially for smaller weapons such as knives and pistols.

TABEL 7
DETECTION STABILITY ACROSS DIFFERENT DISTANCES

| Class | 50cm | 100cm | 150cm | 200cm | 250cm | 300cm | 350cm | 400cm | 450cm | 500cm |
|---|---|---|---|---|---|---|---|---|---|---|
| Sickle | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Machete | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Axe | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ~ |
| Sword | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ~ |
| Knife | ✓ | ✓ | ✓ | ✓ | ~ | ~ | ✗ | ✗ | ✗ | ✗ |
| Pistol | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ~ | ~ |
| Rifle | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Based on Table 7, the detection performance varies significantly across weapon classes, primarily influenced by object size and visual distinctiveness.Sickle, Machete, and Rifle achieved the best performance. All three classes were detected consistently (✓) at every distance from 50 cm to 500 cm. The large surface area and distinctive shapes of these weapons allow the YOLOv8 model to maintain high confidence even at maximum range. Axe and Sword were detected consistently up to 450 cm. At 500 cm, detection became unstable (~), meaning the bounding box appeared intermittently. This instability is likely due to the elongated blade of the sword and the relatively thin axe head, which occupy fewer pixels at long distances. Nevertheless, both classes remained detectable in most frames.

Pistol showed stable detection (✓) from 50 cm to 400 cm. At 450 cm and 500 cm, detection became unstable (~). The pistol's compact size and occasional low contrast against dark clothing or backgrounds caused the confidence score to fluctuate near the threshold. Despite this, the system still recognized the pistol intermittently, which may be sufficient to trigger an alert. Knife exhibited the poorest performance. It was detected consistently only up to 200 cm. At 250–300 cm, detection became unstable (~), and beyond 350 cm, the system completely failed to detect the knife (✗). The knife's small size (often less than 100 pixels in height at 350 cm) and thin profile make feature extraction extremely challenging. Additionally, knives are often held in ways that align the blade with the camera's line of sight, further reducing visible area. Overall, the field testing confirms that object size is the primary factor limiting detection range. Large weapons (rifles, machetes, sickles) can be reliably detected up to 5 meters, while small weapons (knives) require a maximum distance of 2–2.5 meters for reliable detection. These findings provide practical guidance for camera placement in real-world deployments: cameras should be spaced such that no point in the monitored area is farther than 3 meters from a camera to ensure detection of all weapon types, or 2 meters if knife detection is critical.

To determine the real-time detection performance for each object class, the percentage of detection outcomes was calculated based on three categories: detected, unstable detection, and not detected. This calculation aims to describe the system's success rate under various testing conditions. The percentage for each category was calculated using the following equations:

$$\text{Detection Percentage} = \frac{Number\ of\ stable\ detections}{Total\ tests} \text{ x } 100$$

$$\text{Unstable Percentage} = \frac{Number\ of\ unstable\ detections}{Total\ tests} \text{ x } 100$$

$$\text{Not Detected Percentage} = \frac{Number\ of\ failed\ detections}{Total\ tests} \text{ x } 100$$

For example, the knife class had 4 stable detections, 2 unstable detections, and 4 failed detections out of 10 total tests. The percentages are calculated as follows:

$$\text{Detection Percentage} = \frac{4}{10} \text{ x } 100 = 40\%$$

$$\text{Unstable Percentage} = \frac{2}{10} \text{ x } 100 = 20\%$$

$$\text{Not Detected Percentage} = \frac{4}{10} \text{ x } 100 = 40\%$$

These results indicate that detection for the knife class remains inconsistent across the tested conditions. The same calculation was applied to all weapon classes, and the results are presented in Table 8.

TABEL 8
DETECTION PERCENTAGE RESULTS BASED ON DISTANCE TESTING

| Object | Total Tests | Detected (%) | Unstable (%) | Not Detected (%) |
|---|---|---|---|---|
| Sickle | 10 | 100 | 0 | 0 |
| Machete | 10 | 100 | 0 | 0 |
| Axe | 10 | 90 | 10 | 0 |
| Sword | 10 | 90 | 10 | 0 |
| Knife | 10 | 40 | 20 | 40 |
| Pistol | 10 | 80 | 20 | 0 |
| Rifle | 10 | 100 | 0 | 0 |

These findings provide practical guidance for real-world deployment. For campus corridors, cameras should be spaced so that no point in the monitored area is farther than 3 meters from a camera to ensure detection of all weapon types, or 2 meters if knife detection is a priority. Future work should focus on improving small-object detection through higher-resolution input or dedicated network heads.

### *D. Evaluation under Real-World Surveillance Conditions*

Comparison with Previous StudiesThe evaluation results demonstrate that the proposed system remained capable of detecting weapon objects under all tested surveillance conditions. Detection confidence decreased under low illumination, oblique viewing angles, and partial occlusion because fewer visual features were available for extraction. Crowded scenes also slightly reduced detection stability due to object overlap. Nevertheless, all scenarios still produced valid detections with confidence scores above 0.50, indicating that the proposed system maintains acceptable robustness for practical surveillance applications.

TABEL 9
WEAPON DETECTION UNDER REAL-WORLD SURVEILLANCE CONDITIONS

| Scenario | Condition | Confidence | FPS |
|---|---|---|---|
| (a) low lighting | Low-light indoor | 0.75 | 4 |
| (b) the angle of the image | Angle -45 | 0.66 | 4 |
| (c) Occlusion | Partially closed | 0.52 | 3 |
| (d) crowding (high density) | Many people | 0.57 | 4 |

*Note: testing used 1080p video on an Intel Core i3-8130U laptop without a dedicated GPU.*

The evaluation results indicate that the proposed system remained capable of detecting weapons under various real-world surveillance conditions, including low-light environments, oblique viewing angles, partial occlusion, and crowded scenes. Although detection confidence decreased under more challenging scenarios, all test cases still produced valid detections with confidence scores above the predefined threshold. These findings demonstrate that the proposed system maintains acceptable robustness and reliability for practical real-time surveillance applications.

### *E. False Alarm Analysis*

Figure 12. *Broom handle*

Figure 13. *Hot glue gun*

*Figure 14. Working tools*

TABEL
FALSE ALARM EVALUATION USING NON-WEAPON OBJECTS

| Tested Object | Object Category | Bounding Box | Alarm | Detection Result |
|---|---|---|---|---|
| Broom handle | Household tool | No | No | Correctly ignored |
| Hot glue gun | Working tool | No | No | Correctly ignored |
| Working tools | Equipment | No | No | Correctly ignored |

To assess the possibility of false alarms, several non-weapon objects with shapes similar to weapons, including a broom handle, a hot glue gun, and other working tools, were tested. The experimental results showed that none of these objects produced a bounding box, confidence score, or alarm activation, indicating that the proposed YOLOv8 model was able to distinguish weapon objects from visually similar non-weapon objects under the tested conditions. Nevertheless, this evaluation was limited to a small number of representative objects, and future work should include a broader range of similar objects to further validate the system's robustness against false positives.

Since the proposed system automatically captures facial images after weapon detection, privacy and ethical considerations should also be addressed. The captured facial images are intended solely as supporting evidence for security monitoring and incident investigation and should only be accessible to authorized personnel. Furthermore, the stored data should comply with applicable personal data protection regulations and should not be used for purposes unrelated to public security.

## IV. CONCLUSION

This research successfully implemented a web-based real-time detection system for sharp weapons and firearms using the YOLOv8 algorithm. The system is capable of recognizing seven hazardous object classes simultaneously: sickles, machetes, axes, swords, knives, pistols, and rifles. Beyond detection, the system integrates proactive features including audible alarms, real-time detection logs, statistical graphs, and automatic face capture upon weapon detection, transforming passive surveillance into an intelligent early warning system.

The YOLOv8 model was trained on a dataset of 11,445 images, augmented to 27,687 images. Evaluation results showed strong performance with precision of 94.3%, recall of 87.8%, mAP@0.5 of 93.2%, and mAP@0.5:0.95 of 60.1%. Per-class analysis revealed that axes, machetes, and sickles achieved the highest detection accuracy, while knives and pistols showed relatively lower performance due to their smaller size and visual similarity to benign objects. The confusion matrix confirmed that most misclassifications occurred between pistols and knives.

Field testing demonstrated that the system reliably detects large weapons (rifles, machetes, sickles) up to 5 meters, while knives require a closer range of 2–3 meters for stable detection. The web-based dashboard operates at 25–30 FPS on standard hardware, making it suitable for real-world deployment. Despite these promising results, the proposed system still has several limitations. The current evaluation was conducted using a limited number of surveillance scenarios, and the Haar Cascade detector may experience reduced performance under severe occlusion, extreme illumination, or non-frontal face orientations. Moreover, the current implementation records all detected faces without explicitly associating each face with a specific detected weapon. Future work should focus on improving small-object detection through higher-resolution input or dedicated network heads, expanding the dataset with more varied lighting and occlusion conditions, and deploying the system on edge devices for cost-effective campus-wide surveillance.

## REFERENCES


[1] Kepolisian Republik Indonesia, "Kejahatan di Indonesia Selama 2024 Capai 325.150 Kasus," 2024.

[2] Databoks, "Kasus Kejahatan yang Dilaporkan ke Polri Bertambah pada 2025." [Online]. Available:https://databoks.katadata.co.id/demografi/statistik/695b5a6cb6de5/kasus-kejahatan-yang-dilaporkan-ke-polri-bertambah-pada-2025

[3] R. Indonesia, "Undang-Undang Darurat Republik Indonesia Nomor 12 Tahun 1951," vol. 1936, no. 170, pp. 2–4, 1951.

[4] INTR-O Realita, "Dilema Realitas Ruang Aman Kampus." [Online]. Available: https://introrealita.com/dilema-realitas-ruang-aman-kampus/

[5] I. Maulana, N. Rahaningsih, and T. Suprapti, "Analisis penggunaan model YOLOv8 (You Only Look Once) terhadap deteksi citra senjata berbahaya," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3621–3627, 2023.

[6] M. Amelia, "Polda Riau Ungkap Motif Dendam di Balik Pembacokan Mahasiswi UIN Suska," *DetikNews*, 2026, [Online]. Available: https://news.detik.com/melindungi-tuah-marwah/d-8373785/polda-riau-ungkap-motif-dendam-di-balik-pembacokan-mahasiswi-uin-suska

[7] T. Nur, H. Huzaeni, and M. Khadafi, "Implementasi Metode Object Detection Dengan Algoritma You Only Look Once (YOLO) Untuk Deteksi Kecurangan Di Dalam Ruang Ujian," *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 6, no. 1, 2023, doi: 10.30811/jtrik.v6i1.4699.

[8] A. Herlangga, "Penerapan Transfer Learning Efficientnetb3 Untuk Pengenalan Senjata Tradisional Sumatera Barat Menggunakan Convolutional Neural Network (CNN)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4256.

[9] S. Setyawan and E. P. Widiyanto, "Analisis Kinerja Model Yolov8 Untuk Monitoring Kepatuhan Penggunaan Sepatu Safety Pada Petugas Pemadam Kebakaran," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, 2025, doi: 10.23960/jitet.v13i3s1.8065.

[10] H. Supriyanto, S. C. Abadi, and A. Shalsabilah, "Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 1–8, 2024, doi: 10.52158/jacost.v5i1.637.

[11] L. Mahdiyah, S. Oktamuliani, and W. L. Putri, "Penerapan Algoritma Deep Learning YOLOv8 pada Platform Roboflow untuk Segmentasi Citra Panoramik," *Jurnal Fisika Unand*, vol. 14, no. 3, pp. 228–234, 2025, doi: 10.25077/jfu.14.3.228-234.2025.

[12] A. Ramdan and A. Asriyanik, "Implementasi Deteksi Objek Real-Time Sebagai Media Edukasi dengan Algoritma YOLOv8 pada Objek Sampah," *Jurnal SAINTEKOM*, vol. 14, no. 2, pp. 142–153, 2024, doi: 10.33020/saintekom.v14i2.638.

[13] I. C. Pradana, E. Mulyanto, and R. F. Rachmadi, "Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2," *Jurnal Teknik ITS*, vol. 11, no. 2, 2022, doi: 10.12962/j23373539.v11i2.86587.

[14] M. A. Rahman and H. Setiawan, "Helmet and License Plate Detection Using YOLOv8 in Parking Areas." Universitas Muhammadiyah Sidoarjo, 2025. doi: 10.21070/ups.8409.

[15] D. Lasmana, N. L. Husni, and R. D. Kusumanto, "Deteksi Objek Menggunakan YOLOv5 dan YOLOv8 pada Perangkat Pemantauan Lingkungan," vol. 12, pp. 276–282, 2025.

[16] R. Pusparina A and R. Rahmadewi, "Deteksi Objek Berbasis Yolov8 Untuk Mendukung Keselamatan Kerja Di Lokasi Konstruksi," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 3188–3195, 2025, doi: 10.36040/jati.v9i2.13257.

[17] H. Achmad, A. Pramudwiatmoko, M. S. Gumilang, B. Al Karim, and H. Wiyono, "Analisis Kinerja Model Deteksi Objek Yolo, Ssd, dan Faster R-Cnn pada Citra Penglihatan Malam untuk Pengenalan Tindak Kejahatan," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 1, pp. 145–152, 2025.

[18] M. Fikry and Y. Afrillia, "Machine Learning Algorithms Comparison for Gender Identification," *Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS)*, vol. 4, 2024, doi: 10.29103/micoms.v4.2024.