Real-Time Weapon Detection based on Optimized YOLOv8

NOPPAKUN BOONSIM
Department of Technology and Engineering,
Faculty of Interdisciplinary Studies, Khon Kaen University,
112 Khon Kaen University, Nong Khai campus, Nong Khai,
THAILAND

Abstract: - Shootings in schools, shopping malls, and other public places are increasing every year, partly due to the ease of legally acquiring firearms in some nations. This research proposes an automated weapons detection system utilizing optimized transfer learning with the YOLOv8 architecture to analyze CCTV footage in public spaces. The study uses a modified YOLOv8 model with pre-trained weight for real-time object identification to improve security response capabilities. Experiments reveal that the optimized YOLOv8m configuration has a higher detection accuracy and processing efficiency. The model achieved an F1 score of 91.9%, a mean average precision of 94.2%, and a processing speed of 35.7 frames per second using Stochastic Gradient Descent with a momentum of 0.9 and a learning rate of 0.001. The findings show that the proposed approach identifies weapons successfully in real-world public situations, proving its potential to considerably improve public security.

Key-Words: - Real-Time, Weapon, Detection, YOLOv8, CCTV, Hyperparameter, Tuning.

Received: March 13, 2025. Revised: July 11, 2025. Accepted: August 7, 2025. Published: October 16, 2025.

1 Introduction

Over the last decade, more than 40,000 mass shootings have happened in the United States, killing 200,000 people, [1]. Mass shootings in the nation are a complex topic with many underlying causes. Public places such as schools, colleges, government buildings, malls, theaters, and even places of worship have been the scene of these crimes. In 2022, there were more than 400 shootings in total, which unfortunately killed over 40,000 lives, according to [1].

The accessibility of handgun acquisition, along with lax gun control rules, has considerably aggravated the situation. Furthermore, limited access to mental health services, combined with societal factors such as racial, religious, and gender prejudice, has developed a violent society. Effectively addressing the urgent threat of gun violence demands a comprehensive plan combining stricter gun laws (e.g., red flag laws, assault weapon bans, universal checks), improved mental health resources and societal empathy to prevent violence, and technological solutions like computer vision for proactive firearm detection and alerts.

This study presents transfer learning with the YOLOv8 architecture for potential real-time weapon detection in public places. Additionally, optimal parameters are offered, which are determined through techniques targeted at maximizing the accuracy and computing efficiency of the resulting models.

2 Literature Review

Computer image processing research for weapon identification has discovered two key approaches: machine learning and deep learning. Machine learning (ML) techniques for weapon identification often include feature engineering and the use of ML algorithms such as logistic regression (LR), random forests (RF), support vector machines (SVM), and multilayer perceptrons (MLP). While these algorithms achieve excellent accuracy with minimal data and training time, they frequently rely on expert knowledge for feature selection and extraction. Many studies have been conducted to study various machine learning-based methodologies. For example, a study in [2] proposed employing Haar Cascade features for detecting various weapon kinds, achieving a 46% detection accuracy on a 1,560-image dataset. Similarly, Active Appearance Models for detecting knives in images were given, as proposed by [3]. This method used the Harris corner detector to detect knife information. The approach was tested on only 40 photos and demonstrated an accuracy of approximately 92.50%. Although the accuracy was great, the testing sample size was tiny.

Furthermore, in 2015 a visual gun detection system was introduced that uses color and a k-means clustering algorithm to detect gun candidate objects. Specifically, the Harris interest points detector and the FREAK description are used to determine whether a detected blob corresponds to a gun. The accuracy was 83.07% on a dataset of 65 images, [4]. In addition, during the same year research team [5] introduced the SURF feature to handle the problem of recognizing firearms in images with low light, noise interference, and images acquired from various camera angles. All the research listed above conducted experiments using private datasets and primarily static photographs, which are unsuitable for CCTV work that requires moving and long-range images.

The second technique, deep learning, employs neural networks with multiple layers to learn features directly from training images, eliminating the need for manual feature engineering. This approach generally outperforms traditional machine learning methods but requires substantial training data and time, often several weeks. Deep learning techniques can be further categorized into one-step and two-step methods. The first sub-technique, one-stage deep object detection, processes images directly to detect objects in a single step. This category encompasses Single Shot Detection (SSD) and You Only Look Once (YOLO). An investigation into weapon detection utilizing SSD and Faster in [6] was conducted on both pre-labeled and self-created datasets. While both algorithms were practical, real-time applications demand a balance of speed and precision. SSD outperformed in speed but lacked accuracy when compared to Faster RCNN. The research team [7] solved the issue of false negatives in video surveillance weapon identification by offering a novel top-down approach using human poses. Unlike previous studies, which focused on specific weapon types, this method uses human position estimates to guide the detection of hand areas, which a weapon detection model then examines. The Adaptive Pose Factor is a novel feature that optimizes hand regional localization based on the distance between the body and the camera. In indoor and outdoor surveillance scenarios, the proposed Weapon Detection over Pose Estimation (WeDePE) approach outperforms bottom-up approaches and existing detection algorithms. The Sohas weapon dataset was used to test four approaches: FRCNN, SSD, EfficientDet, and CenterNet. EfficientDet has a higher precision score than the other models, at 94.4.

Several research groups have investigated using YOLO-based object detection for weapon detection. A study in [8] examined the growing demand for firearms as crime rates rise, emphasizing the need for novel solutions to promote public safety. The authors provided a method for detecting guns in photographs using convolutional neural networks (CNNs), specifically the YOLO (You Only Look Once) object detector. They accelerated the learning process by employing YOLO to annotate an unmarked image database, resulting in a new integrated database. The suggested method was evaluated using a dataset of 608 photographs, 304 of which contain weapons. The experimental results demonstrated a sensitivity of 100.00% and an accuracy of 89.15%. Next, a two-stage system concept: a "Front End" to detect people and a "Back End" to identify firearms near them is proposed to solve the limitations of closed-circuit television (CCTV) monitoring systems in responding quickly to risky scenarios such as armed robbery, particularly when observed by human operators, [9]. The performance of the firearm detection system is examined using several convolutional neural network (CNN) architectures in [10], with metrics such as recall and precision reaching up to 86% in a network configuration based on VGG Net and grayscale images. A refined technique was developed by employing YOLO V3 on a custom dataset to recognize handguns and rifles. Their training results reveal that YOLO V3 outperforms YOLO V2 and standard CNNs. The approach does not necessitate extensive GPU resources because transfer learning is used. By incorporating this approach into surveillance systems, the authors hope to improve public safety and potentially reduce incidences of manslaughter and mass killings.

Furthermore, they believe that the technology might be used by high-end surveillance robots to detect weapons or other dangers, reducing the risk to human life. The accuracy on a private dataset was reported at 98.89%. In [11], a comparative evaluation was conducted on several algorithms, including VGG16, Inception-V3, Inception-ResnetV2, SSDMobileNetV1, Faster-RCNN Inception-ResnetV2 (FRIRv2), YOLOv3, and YOLOv4. The experiment revealed that Yolov4 was the top performance, with an F1-score of 91% and a mean average precision of 91.73%. Finally, a combined technique of YOLO and Area of Interest (AOI) in [12] that uses preprocessed frames and Gaussian blur is proposed to remove the background. According to the research findings, YOLO-v5 has high recall rates and fast detection

speeds, making it an ideal choice for security and weapon detection applications.

Two-stage deep object detection is the second sub-technique. It consists of two steps: first, it locates items in an image; second, it classifies the objects it finds and determines their locations. This approach commonly uses Faster Region-based Convolutional Neural Networks (FRCNN). First, [13] employed Faster R-CNN to investigate the prevalence of portable firearms in criminal behavior. To accomplish this, the researchers employed a pre-trained DCN model from the Faster R-CNN architecture. The proposed method was evaluated on the Internet Movie Guns Database (IMFDB), a benchmark gun database, and yielded good results for identifying visual portable guns. The experiment achieved a classification accuracy of 92.60% on a specific, private dataset. However, the results may not apply to other, different datasets. Next, a two-step approach for detecting pistols in surveillance films was proposed in [14]. A quicker R-CNN model trained on the new database outperformed lower-quality YouTube videos. The system correctly raised alarms in 27 of 30 scenarios, detecting five consecutive true positives in 0.2 seconds. The authors developed a new statistic called Alarm Activation Time per Interval (AATpI) to assess detection model performance. On a publicly available dataset of 3,000 images, the system obtained 91.43% accuracy. However, it has difficulty detecting firearms in images with low contrast and blurriness. However, the system's effectiveness in difficult environments, such as low light or crowded settings, warrants additional examination. Then, research group [15] focused on active shooter detection. The researchers used feature extraction techniques and a CNN classifier to distinguish guns from other things, with a classification accuracy of 97.78%. However, the model's performance in real-world scenarios requires validation. In [16], a technique called DaCoLT was introduced to improve detection performance under varying lighting conditions. The constructed model performed exceptionally well, with Faster R-CNN (Inception-ResNet-V2) achieving an F1 score of 95%. However, its inference speed of 1.3 frames per second (FPS) limited its suitability for near-real-time applications. A two-tiered approach was advised to detect small, hand-manipulated objects in video surveillance [17]. The first level chose candidate regions, and the second used binarization techniques with CNN classifiers. The study focused on spotting weapons and objects that resemble firearms or knives, such as pistols, knives, smartphones, money, handbags, and credit cards. The proposed methodology successfully reduced false positives compared to baseline multi-class detection models. Synthetic datasets, a promising approach to augment training data and improve model performance, were employed in [18]. The study in [19] applied Faster R-CNN with Feature Pyramid Network using ResNet-50 to detect weapons in real-time CCTV applications. The generated datasets were made available to the research community. However, the quality and diversity of synthetic data are crucial factors. The Weakly Supervised Orientation Aware Object Detection (OAOD) technique was initially introduced in 2021 and uses training data annotated with axis-aligned bounding boxes (AABB) to detect oriented object bounding boxes. This is achieved using a multistage technique, with Stage 1 predicting AABB and Stage 2 predicting OBB. An oriented proposal generation module and object-aligned Region of Interest (RoI) pooling extract features depending on expected orientation, resulting in orientation invariance. The approach was assessed on a private dataset and compared with state-of-the-art object detectors. It outperformed both fully supervised oriented object detectors by a wide margin, achieving a mean average precision (mAP) of 88.3 on AABB and 77.5 on OBB tested on 10,000 images. The same year, [20] proposed a deep learning method built on the VGGNet architecture to detect seven different types of weapons. The model was trained to recognize hunting rifles, knives, handguns, revolvers, bazookas, grenades, and assault firearms. For training, a new dataset was created that included all seven types of weapons. When compared to VGG-16, ResNet-50, and ResNet-101, the suggested model's classification performance was assessed; at 98.40%, the suggested method provided the highest success rate. Finally, a brightness-boosting technique is proposed in [21] to improve knife detection in low-light conditions. Experimental results confirm the efficacy of the suggested technique, which employs deep transfer learning methods. Furthermore, the study compares the performance of four prominent deep convolutional neural network models to determine which one is better for knife detection. Finally, a comparison with state-of-the-art techniques shows that the suggested technique is superior in knife identification, with Alexnet yielding the highest accuracy of 96.95%.

In terms of accuracy, Faster R-CNN surpasses one-stage methods like SSD and YOLO since it detects regions of interest (ROIs) before classifying them into a predefined object class. However, the

computational complexity of Faster R-CNN could make it inappropriate for real-time surveillance applications in CCTV cameras due to slower inference times. This work proposes an enhanced YOLOv8 model for weapon identification to address this issue. This approach uses sophisticated techniques like data augmentation, label smoothing, enhanced loss functions, a strong anchor box design, and decoupled head and label assignment to improve inference speed and accuracy. Because of the trade-off between processing time and detection accuracy, the optimized YOLOv8 model could be helpful in real-time surveillance systems.

3 Methodology

This study proposes YOLOv8 as a potential method for real-time weapon identification in CCTV systems. For this application, YOLOv8 was chosen due to its significantly higher operational speed and accuracy when compared to SSD and Faster R-CNN approaches. YOLOv8 [22] is a sophisticated object detection architecture that employs a single-stage, anchor-free technique. It uses a deep convolutional neural network (CNN) with a backbone and neck network. The backbone network extracts features from the input image, while the neck network incorporates these features at various scales. The network's final layer predicts bounding boxes and class probabilities for all objects in the image. Advanced methods, including EfficientNet as the backbone, a BiFPN neck for effective feature integration, and a unique head design for increased detection accuracy, are all incorporated into YOLOv8.

Figure 1 (Appendix) depicts an approach for detecting weapons in images. It employs transfer learning from the YOLOv8 paradigm. The technique takes training images of weapons and their positions, transmits them to the YOLOv8 model, and then trains it. The parameters that are adjusted to attain the maximum accuracy include the optimization technique (e.g., "SGD," "Adam," and "AdamW"), model learning rate, momentum, and model size, as indicated in Table 1.

This method provides benefits such as faster training, more efficient object identification, and the effective separation of regression and classification algorithms. Hyperparameter tuning was performed to improve further the model for optimal precision and real-time weapon identification in CCTV images. Table 1 shows the specific hyperparameters used during the tuning procedure. The model will be trained and tested using photos with three resolutions: 320x320, 416x416, and 640x640 pixels. Three algorithms will be investigated to optimize the training process: Stochastic Gradient Descent (SGD), Adam, and AdamW. A variety of learning rates (1e-5 to 1e-1) and momentum values (0.5 to 0.9) will be thoroughly assessed. Additionally, three pre-trained model weights (micro, small, and medium) will be used as initializations to speed up convergence and improve performance.

Table 1. Hyperparameters search space

Parameter	Search Space
Image size	320, 416, 640
Optimizer	'SGD', 'Adam', 'AdamW'
Learning rate	e-5, e-4, e-3, e-2, e-1
Momentum	0.5,0.6,0.7,0.8, 0.9
Model weights	nano, small, medium

4 Experiment Results and Discussion

Three studies were performed to compare YOLOv8 to YOLOv5 and YOLOv6. The experiment compared the efficacy of YOLOv8's dual-head detection architecture to YOLOv5, a single-head model, and YOLOv6, another dual-head model with a unique network backbone structure. The models were trained on datasets with three different image sizes: 320, 416, and 640 pixels. Three architectural versions were examined to see how model complexity affected performance: nano, small, and medium. The present study limited its focus to these three networks owing to their smaller size and lower resource consumption, which may render them more suitable for real-time applications.

A publicly available dataset from Olmos et al. [14] was employed for experimental evaluation. This dataset consists of 12,869 images categorized into 102 distinct classes. Our research focused on developing a model capable of accurately detecting weapons, particularly handguns and knives, prevalent in urban crime scenes. Therefore, a subset of this dataset was curated, comprising 3,000 images of guns and 2,057 images of knives. Two thousand images were chosen randomly from each category, yielding 4,000 images for our trials, as shown in Figure 2(a-d). This curated dataset underwent extensive data augmentation to enhance its diversity and robustness. Augmentation techniques included HSV color adjustments, geometric transformations, horizontal and vertical flipping, and image mixing.

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

Fig. 2: Example of training images

All experiments were performed on a Windows 10 system with an Intel Core i7-10700 processor (2.9 GHz), 32 GB DDR4 RAM, and a single NVIDIA GeForce GTX 1080 graphics card. Python version 3.10, Pytorch version 2.2.1, and Cuda version 11.8 were the programming language and libraries.

This study uses the following measurements to assess the technique's efficiency: accuracy, precision, recall, and F1 score, as stated in Equations 1–4.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (4)$$

A true positive (TP) occurs when the model accurately identifies an object within the image, aligning with the ground truth. False Positive arises when the model erroneously predicts the presence of an object in a region where none exists. False Negative occurs when the model fails to detect an object that is indeed present within the image. A True Negative occurs when the model correctly predicts the absence of an object in a region where no object is present. As demonstrated in equation 1, accuracy indicates a model's overall ability to classify objects within images correctly. It is measured as the proportion of correct predictions (true positives and true negatives) to the total number of forecasts. Precision is a statistic that evaluates the expected positive outcome of predicted observations within a positive class, as seen in Equation 2.

The recall is calculated as the number of correctly detected positive observations in Equation 3. The F1-Score score, illustrated in Equation 4, shows the harmonic weight of precision and recall.

4.1 Experiment 1

A single-headed YOLOv5 architecture served as the foundational model for this experiment. Transfer learning was employed to adapt this model for weapon detection in images. Table 2 (Appendix) provides a summary of the experimental results. The abbreviations TP, FP, FN, Pre, Rec, Acc, F1, and mAP represent True Positive, False Positive, False Negative, Precision, Recall, Accuracy, F1 Score, and Mean Average Precision, respectively. Through rigorous hyperparameter tuning, the optimal configuration was determined to be a medium-weight model with an input image size of 640 pixels, the SGD optimizer with a learning rate of 0.001, and a momentum of 0.9. Under these conditions, the model achieved a precision of 90.3%, a recall of 88.0%, an F1 score of 89.1%, and a mean Average Precision of 91.7%.

Table 3 (Appendix) summarizes the performance comparison between GPU- and CPU-based weapon detection. The results highlight the substantial speed advantage of GPU-accelerated models. The study demonstrates that GPU-based detection consistently achieves real-time performance on CCTV systems, processing images at a rate exceeding 30 frames per second. In contrast, CPU-based detection generally failed to meet the temporal constraint. Notably, nano models trained on reduced image dimensions (320 and 416 pixels) demonstrated comparable efficacy on CPUs, attaining mean average precision (mAP) scores of 76.6% and 80.4%, respectively.

Although the YOLOv5 medium model demonstrated superior mAP of over 90% when trained on 640-pixel images, its computational demands on a CPU, resulting in an inference time of 2.5 FPS, rendered it unsuitable for real-time applications. The smaller nano and small models exhibited faster CPU performance (greater than 15 FPS) but compromised accuracy, achieving mAP of approximately 80%. However, the medium model's GPU-accelerated inference speed of 27.8 FPS makes it a viable candidate for real-time CCTV applications.

4.2 Experiment 2

Experiment 2 followed a similar procedure as Experiment 1 but utilized the YOLOv6 model, which features a decouple-head architecture for object detection: one head for weapon classification and the other for weapon localization. As shown in Table 4 (Appendix), the model trained on 640-pixel images using a medium network architecture exhibited the best performance, achieved a precision of 90.8%, a recall of 88.0%, an F1 score of 89.4 %, and a mean average precision of 93.1% with a learning rate of 0.0001, momentum of 0.8, and the Adam optimizer.

Table 5 (Appendix) presents a comparative analysis of the performance of YOLOv6 models across varying image sizes. The study evaluated three model variants: nano, small, and medium. The table provides training and testing image sizes for each model and inference latency measurements on CPU and GPU platforms.

Among the models evaluated, the YOLOv6 medium model, trained on 640-pixel images, exhibited the highest performance, achieving a mean average precision of 93.1%.

While its GPU-accelerated inference speed of 29.5 frames per second (FPS) is suitable for real-time closed-circuit television (CCTV) applications, its CPU performance of 2.6 FPS is inadequate. Although the small model demonstrated faster CPU performance (exceeding 15 FPS), it exhibited lower accuracy levels, with a mean average precision of approximately 82%.

4.3 Experiment 3

In experiment 3, the YOLOv8 model was evaluated with various image sizes and network structures. The findings presented in Table 6 (Appendix) demonstrate that the medium-sized network trained on 640-pixel images yielded the best results, achieving a precision of 94.8%, a recall of 89.1%, F1 score of 91.9%, and a mean average precision of 94.0% when employing a stochastic gradient descent (SGD) optimizer with a learning rate of 0.001 and a momentum of 0.9.

Table 7 (Appendix) analyzes the performance of YOLOv8 models across various image sizes. It evaluates nano, small, and medium models, providing training and testing data and inference latency on CPU and GPU.

Experimental results indicate that the YOLOv8 medium model, trained on 640-pixel images, achieved the highest performance, with a 91.9 % F1 score rate and a mean average precision of 94.2%. When evaluated on a Graphics Processing Unit (GPU), the medium-sized model demonstrated exceptional real-time capabilities, processing images at 35 frames per second (FPS).

Furthermore, across all YOLOv8 architectures, the average frame rate exceeded 85 frames per second, outperforming the performance of YOLOv5 and YOLOv6. However, when operating on a Central Processing Unit (CPU), the average frame rate decreased to approximately 17 frames per second, making it unsuitable for real-time applications. When trained on smaller image sizes (416 and 320 pixels), the nano and small architecture exhibited acceptable CPU performance (over 15 FPS) while maintaining a reasonable mAP of approximately 85%. Therefore, for applications constrained to CPU processing, the small architecture with 320-pixel images or the nano architecture with 416 or 640-pixel images could be viable options.

Figure 3 (Appendix) presents a comparative analysis of three state-of-the-art object detection models: YOLOv5, YOLOv6, and YOLOv8. The models were evaluated across multiple metrics: precision, recall, F1-score, accuracy, and mean Average Precision (mAP). Regarding mAP, YOLOv8 demonstrated superior performance, achieving a maximum accuracy of 94.2%. This result represents a 3% improvement over the single-head YOLOv5 and a 1% improvement over the dual-head YOLOv6, which shares a similar architecture with YOLOv8. Similarly, YOLOv8 exhibited the highest F1 score, attaining 91%. This score surpasses both YOLOv5 and YOLOv6 by 2%.

Table 8 (Appendix) presents a comparative analysis of the proposed method with existing research, focusing on key parameters such as methodology, accuracy, processing speed (frames per second, FPS), and dataset type. For the N/A values in the FPS column, it can be inferred that processing time is comparable to the average framework processing time. Studies utilizing private datasets reported accuracies exceeding 95%, but direct comparison is limited due to the inaccessibility of these datasets for testing the proposed technique. The research employed a YOLOv8 single-stage object detector model, achieving an F1 score of 91.9%. While this performance is not the highest reported, it demonstrates a competitive level suitable for practical applications.

In contrast, two-stage detectors like Faster R-CNN, often achieving accuracies over 93%, typically suffer from higher computational costs, limiting their suitability for real-time scenarios. Among single-stage detectors, the proposed model stands out with its highest accuracy and ranks

second in inference speed, following a study [12] that achieved 100 FPS. Despite not being the fastest, the model's speed of 35.7 FPS is considered sufficient for real-time tasks.

5 Conclusion

This study presents a YOLOv8-based method for detecting weapons in images. The goal is to implement this technology in public CCTV systems, schools, universities, and shopping malls to prevent weapon-related incidents. The proposed method was evaluated on a public dataset, achieving a 94% accuracy rate with a processing speed of 35 frames per second, making it suitable for real-time applications.

The proposed method was assessed using a publicly available dataset [14]. Approximately 5,000 images of guns and knives of various sizes that were taken both indoors and outdoors are included in this dataset. Although the dataset offers a suitable foundation for the first assessment, it is limited by illumination and occlusion conditions that are less varied than those observed in real-world situations. Future studies should focus on resolving these issues to increase the model's resilience and flexibility in real-world situations, especially considering this work in real-time. While the current dataset focuses on pistols and knives, future research could consider detecting other weapons, such as machine guns, rifles, shotguns, and bazookas. Expanding the training dataset to more than 5,000 images might enhance the model's performance. Furthermore, studying alternative topologies to the YOLO family may improve smaller objects' accuracy, speed, and identification.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the author used OpenAI's ChatGPT in order to improve readability and language. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References:

[1] Everytown Research & Policy. Gun Violence in America. (2022), [Online]. https://everytownresearch.org/report/gun-violence-in-america/ (Accessed Date: September 3, 2024).

[2] Zywicki, M., Matiolanski, A., Orzechowski, T.M., Dziech, A., "Knife detection as a subset of object detection approach based on Haar cascades," in *Proceedings of 11th International Conference Pattern recognition and information processing*, 2011, pp. 139-142.

[3] Glowacz, A., Kmieć, M., Dziech, A., Visual detection of knives in security applications using Active Appearance Models, *Multimedia Tools and Applications*, Vol. 74, 2015, pp. 4253-4267, https://doi.org/10.1007/s11042-013-1537-2.

[4] Tiwari, R.K.,Verma, G.K., A computer vision based framework for visual gun detection using harris interest point detector, *Procedia Computer Science*, Vol. 54, 2015, pp. 703-712, https://doi.org/10.1016/j.procs.2015.06.083.

[5] Tiwari, R.K., Verma, G.K., "A computer vision based framework for visual gun detection using SURF," in *Proceedings of International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, 2015, pp. 1-5, https://doi.org/10.1109/EESCO.2015.7253863

[6] Jain, H., Vikram, A., Kashyap, A., Jain, A., "Weapon detection using artificial intelligence and deep learning for security applications," in *Proceedings of International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 193-198, https://doi.org/10.1109/ICESC48915.2020.9155832.

[7] Lamas, A., Tabik, S., Montes, A.C., Pérez-Hernández, F., García, J., Olmos, R., Herrera, F., Human pose estimation for mitigating false negatives in weapon detection in video-surveillance, *Neurocomputing*, Vol. 489, 2022, pp. 488-503, https://doi.org/10.1016/j.neucom.2021.12.059.

[8] Cardoso, G.V.S., Ciarelli, P.M., Vassallo, R.F., "Use of deep learning for firearms detection in images," in *Anais do XV Workshop de Visão Computacional*, 2019, pp. 109-114, https://doi.org/10.5753/wvc.2019.7637.

[9] Romero, D., Salamea, C., Convolutional models for the detection of firearms in surveillance videos, *Applied Sciences*, Vol. 9, Issue 15, 2019, pp.2965, https://doi.org/0.3390/app9152965.

[10] Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C., Anjum, M.R., Weapon

detection using YOLO V3 for smart surveillance system, *Mathematical Problems in Engineering*, Vol. 1, 2021, pp. 1-9, https://doi.org/10.1155/2021/9975700.

[11] Bhatti, M.T., Khan, M.G., Aslam, M., Fiaz, M.J., Weapon detection in real-time CCTV videos using deep learning, *IEEE Access*, Vol. 9, 2021, pp. 34366-34382, https://doi.org/10.1109/ACCESS.2021.3059170.

[12] Ashraf, A.H., Imran, M., Qahtani, A.M., Alsufyani, A., Almutiry, O., Mahmood, A., Attique, M., Habib, M., Weapons detection for security and video surveillance using cnn and YOLO-v5s, *Computer Materials & Continua*, Vol. 70, Issue 4, 2022, pp. 2761-2775, https://doi.org/10.32604/cmc.2022.018785.

[13] Verma, G.K., Dhillon, A., "A handheld gun detection using faster r-cnn deep learning," in *Proceedings of 7th international conference on computer and communication technology*, 2017, pp. 84-88, https://doi.org/10.1145/3154979.3154988.

[14] Olmos, R., Tabik, S., Herrera, F., Automatic handgun detection alarm in videos using deep learning, *Neurocomputing*, Vol. 275, 2018, pp. 66-72, https://doi.org/10.1016/j.neucom.2017.05.012.

[15] Gelana, F., Yadav, A., "Firearm detection from surveillance cameras using image processing and machine learning techniques," in *Proceedings of International Conference on Smart Innovations in Communication and Computational Sciences*, 2019, pp. 25-34, https://doi.org/10.1007/978-981-13-2414-7_3.

[16] Castillo, A., Tabik, S., Pérez, F., Olmos, R., Herrera, F., Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning, *Neurocomputing*, Vol. 330, 2019, pp. 151-161, https://doi.org/10.1016/j.neucom.2018.10.076.

[17] Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., Herrera, F., Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance, *Knowledge-Based Systems*, Vol. 194, 2020, pp. 105590, https://doi.org/10.1016/j.knosys.2020.105590.

[18] González, J.L.S., Zaccaro, C., Álvarez-García, J.A., Morillo, L.M.S., Caparrini, F.S., Real-time gun detection in CCTV: An open problem, *Neural networks*, Vol. 132, 2020, pp. 297-308, https://doi.org/10.1016/j.neunet.2020.09.013.

[19] Iqbal, J., Munir, M.A., Mahmood, A., Ali, A.R., Ali, M., Leveraging orientation for weakly supervised object detection with application to firearm localization, *Neurocomputing*, Vol. 440, 2021, pp. 310-320, https://doi.org/10.1016/j.neucom.2021.01.075.

[20] Kaya, V., Tuncer, S., Baran, A., Detection and classification of different weapon types using deep learning, *Applied Sciences*, Vol. 11, Issue 16, 2021, pp.7535, https://doi.org/10.3390/app11167535.

[21] Galab, M.K., Taha, A., Zayed, H.H., Adaptive technique for brightness enhancement of automated knife detection in surveillance video with deep learning, *Arabian Journal for Science and Engineering*, Vol. 46, Issue 4, 2021, pp. 4049-4058, https://doi.org/10.1007/s13369-021-05401-4.

[22] Hussain, M., YOLO-v1 to YOLO-v8 the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection, *Machines*, Vol. 11, Issue 7, 2023, pp. 677, https://doi.org/10.3390/machines11070677.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

This article was solely authored by Noppakun Boonsim. The author confirms that no undisclosed contributions or ghostwriting were involved in its creation.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US

APPENDIX

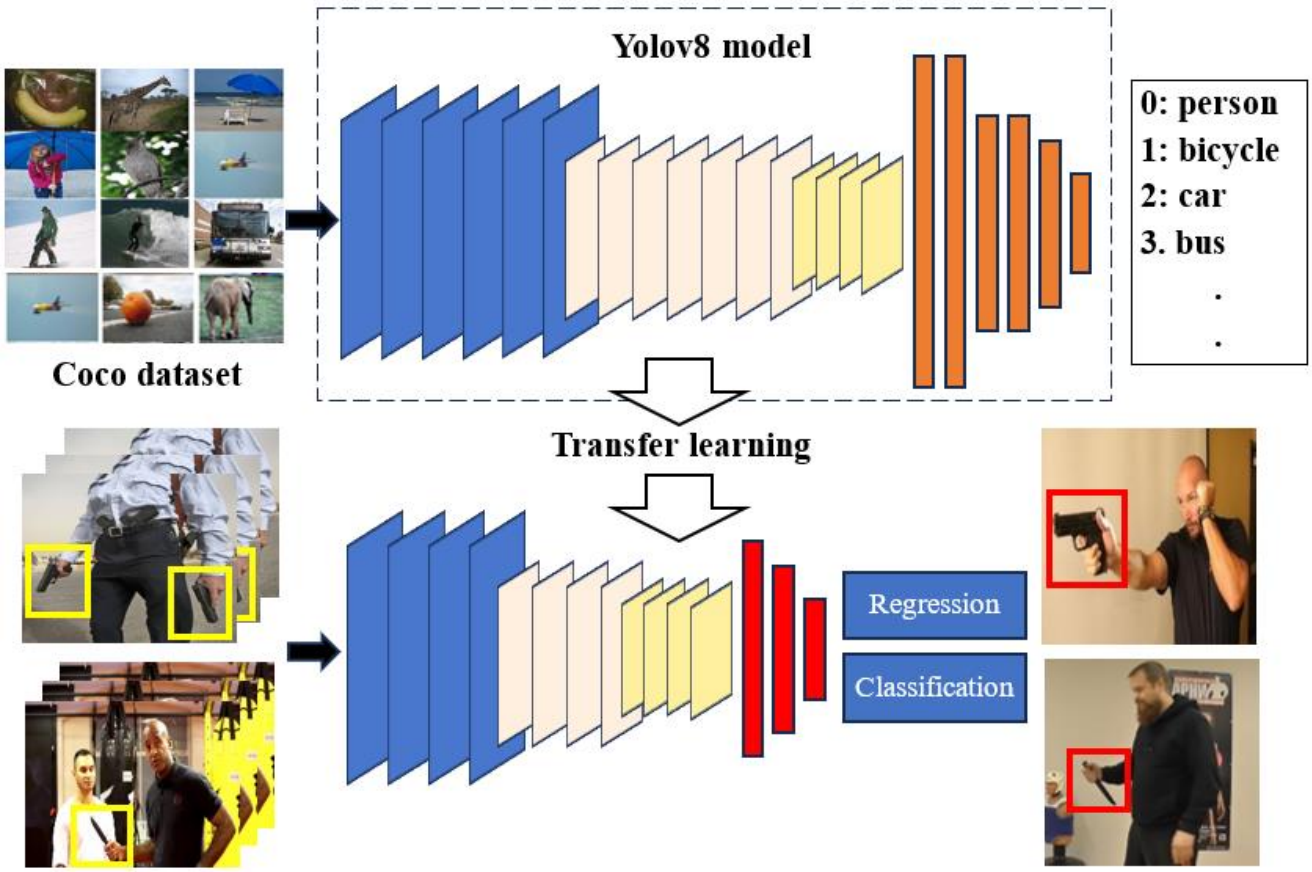

Fig. 1: Overview of the proposed architecture

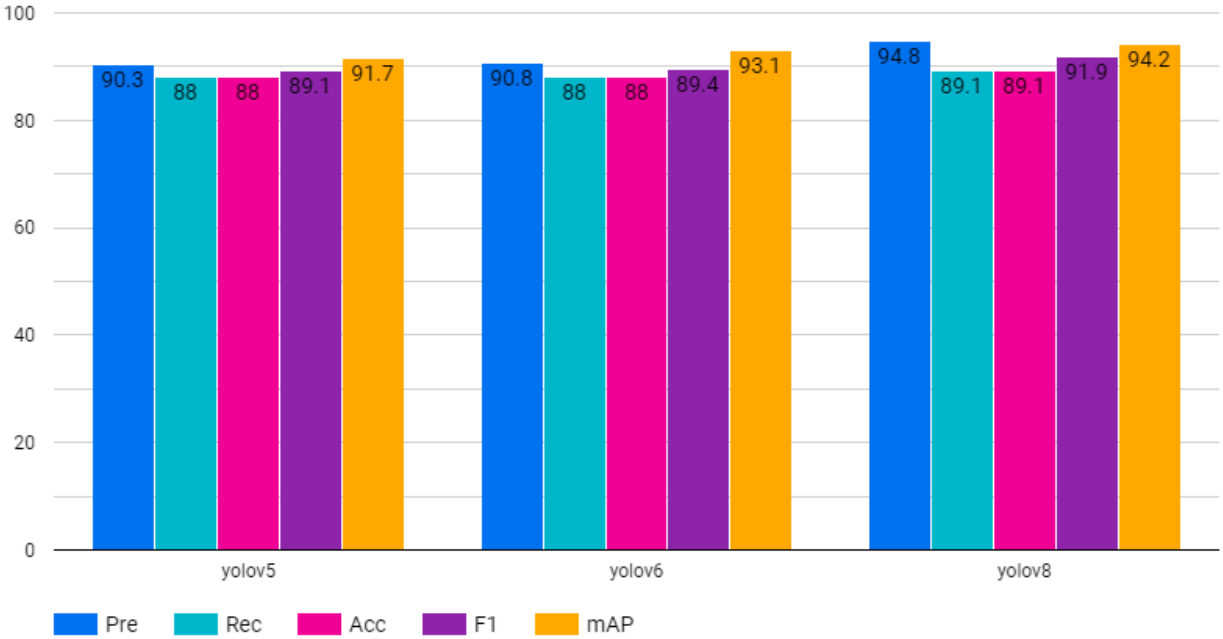

Fig. 3: A comparison of the performance of YOLO models

Table 2. Evaluation of YOLOv5 Models for Weapon Detection

Model	Weights	Size	TP	FP	FN	Pre	Rec	Acc	F1	mAP
Yolov5	Nano	320	641	127	296	83.5	68.4	68.4	75.2	76.6
		416	692	150	245	82.2	73.9	73.9	77.8	80.4
		640	690	115	247	85.7	73.6	73.6	79.2	82.5
	Small	320	696	98	241	87.7	74.3	74.3	80.4	83.2
		416	737	105	200	87.5	78.7	78.7	82.9	86.1
		640	774	97	163	88.9	82.6	82.6	85.6	88.5
	Medium	320	755	77	182	90.8	80.6	80.6	85.4	88.3
		416	770	96	167	88.9	82.2	82.2	85.4	90.1
		640	825	89	112	90.3	88	88.0	89.1	91.7

Table 3. Comparison of Inference Speed for the YOLOv5 Model

Model	Weights	training size	Testing size	CPU (FPS)	GPU (FPS)
Yolov5	Nano	320	320	41.8	35.8
		416	416	30.4	34.6
		640	640	15.7	33.4
	Small	320	320	18.6	37.2
		416	416	13.2	34.6
		640	640	6.2	33.4
	Medium	320	320	8.0	32.7
		416	416	5.2	32.3
		640	640	2.5	27.8

Table 4. Evaluation of YOLOv6 Models for Weapon Detection

Model	Weights	Size	TP	FP	FN	Pre	Rec	Acc	F1	mAP
Yolov6	Nano	320	641	116	296	84.7	68.4	68.4	75.7	76.7
		416	692	103	259	86.8	72.4	72.4	78.9	81.4
		640	690	136	229	83.9	75.6	75.6	79.5	82.5
	Small	320	696	103	246	87	73.7	73.7	79.8	82.6
		416	737	87	234	89	75	75.0	81.4	85.8
		640	774	107	148	88.1	84.2	84.2	86.1	89.2
	Medium	320	755	77	171	90.9	81.7	81.7	86.1	88.1
		416	770	89	148	89.9	84.2	84.2	87.0	91.5
		640	825	84	112	90.8	88	88.0	89.4	93.1

Table 5. Comparison of Inference Speed for the YOLOv6 Model

Model	Weights	training size	Testing size	CPU (FPS)	GPU (FPS)
Yolov6	Nano	320	320	45.7	35.8
		416	416	31.4	35.7
		640	640	15.9	34.6
	Small	320	320	18.9	37.1
		416	416	11.8	34.6
		640	640	6.0	32.3
	Medium	320	320	7.9	31.4
		416	416	5.4	30.7
		640	640	2.6	29.5

Table 6. Evaluation of YOLOv8 Models for Weapon Detection.

Model	Weights	Size	TP	FP	FN	Pre	Rec	Acc	F1	mAP
Yolov8	Nano	320	641	143	296	81.9	68.4	68.4	74.5	77.7
		416	692	56	245	92.7	73.8	73.9	82.2	85.9
		640	739	96	197	86.6	79	78.9	82.6	86.7
	Small	320	739	111	198	87.2	78.9	78.9	82.8	85.9
		416	755	107	182	87.5	80.6	80.6	83.9	88.3
		640	785	91	152	89.7	84.1	83.8	86.8	90.8
	Medium	320	773	92	164	89.2	82.4	82.5	85.7	90.7
		416	797	75	140	91.2	85	85.1	88.0	92
		640	835	44	102	94.8	89.1	89.1	91.9	94.2

Table 7. Comparison of Inference Speed for the YOLOv8 Model

Model	Weights	training size	Testing size	CPU (FPS)	GPU (FPS)
Yolov8	Nano	320	320	28.7	125.0
		416	416	23.3	112.3
		640	640	15.9	100.0
	Small	320	320	18.2	111.1
		416	416	13.0	100.0
		640	640	8.2	83.3
	Medium	320	320	10.2	58.8
		416	416	8.2	43.5
		640	640	3.7	35.7

Table 8. Comparison of Weapon Detection Models

Works	#image	#Class	Method	F1 score	FPS	Dataset
Lamas et al. [7]	17,684	6	SSD	92.9	15	Public
Romero et al. [9]	17,684	1	Yolo	86.0	N/A	Private
Narejo et al. [10]	N/A	2	Yolov3	98.9 (Acc)	N/A	Private
Bhatti et al. [11]	8,327	1	Yolov4	91.7	N/A	Private
Ashrat et al. [12]	3,000	1	Yolov5	89.1	100	Public
Olmos et al. [14]	3,000	1	FasterRCNN	91.43	N/A	Public
Gelana et al. [15]	5,869	1	CNN	97.8 (Acc)	N/A	Private
Pérez-Hernández et al. [17]	5,680	6	FasterRCNN	93.4	0.227	Public
Kaya et al. [20]	5,214	8	FasterRCNN	92.9	0.33	Private
Galeb et al. [21]	12,899	1	FasterRCNN	98.4	N/A	Public
Proposed method	**5,078**	**2**	**Yolov8**	**91.9**	**35.7**	**Public**