Smart Surveillance System for Firearm Detection Using Deep Learning Model

Dr. A.V.S Siva Rama Rao.
Ms. P. Ramya Venkata Lakshmi.
Department of CSE – AIML
Sasi Institute of technology and engineering

Chintha ManiSaiDeep,
Dept of CSE (AI & ML),
Sasi Institute of Technology and Engineering,
Saideep.chintha@sasi.ac.in

Karri ManikantaSai,
Dept of CSE (AI & ML),
Sasi institute of Technology and Engineering,
manikanta.karri@sasi.ac.in

Karri N V Bhaskara Reddy,
Dept of CSE (AI & ML),
Sasi Institute of Technology and Engineering,
bhakarreddy.karri@sasi.ac.in

Ganta Yebbeju Kumar,
Dept of CSE (AI & ML),
Sasi institute of Technology and Engineering,
kumar.ganta@sasi.ac.in

Abstract

Violence in public places is increasing quickly, and a major issue is that people are carrying dangerous weapons like guns and knives in crowded areas without anyone noticing. Security staff cannot watch every camera feed all the time, which often causes delays in responses and raises the risk to public safety. To address this, we developed a Smart Surveillance System that automatically detects weapons in real time and quickly notifies the authorities so that action can be taken before any harm happens.

We used two deep learning models, YOLOv5m and YOLOv8m, training both on the same labelled dataset of 16,000 images that included guns and knives with appropriate bounding box annotations. In testing, YOLOv5m achieved a precision of 80.67%, a recall of 75.72%, and an mAP50 of 81.69%. YOLOv8m performed better with a precision of 87.54%, a recall of 84.08%, and a higher mAP50 of 88.54%. These results show that YOLOv8m is more accurate in detecting weapons, while both models effectively identified guns and knives in real surveillance situations. This system is a practical step toward making public spaces safer by providing automated, continuous, and reliable weapon detection.

Keywords: Weapon Detection, Smart Surveillance, Deep Learning, YOLOv5m, YOLOv8m, Object Detection, Real-Time Detection, Firearm Detection, Knife Detection, Bounding Box Annotation, Public Safety, CCTV Surveillance, Mean Average Precision (mAP), Alert System.

1. Introduction

In recent years, public safety has become a serious concern around the world. Incidents involving weapons like guns and knives in crowded places such as railway stations, bus stands, shopping malls, schools, and streets are increasing at an alarming rate. Often, by the time a security person sees a threat on a camera feed, it is too late to prevent harm. This delay in detection and response is one of the biggest weaknesses in traditional surveillance systems, which rely entirely on human attention and manual monitoring.

Human-based monitoring has clear limits. A security operator cannot watch multiple camera feeds at once without losing focus. Fatigue, distraction, and slow reaction times make it nearly impossible to catch every threat in real time.

There is a strong need for a system that can do this job automatically, without getting tired, and without missing anything.

This is exactly what our project aims to solve. We have developed a Smart Surveillance System that uses the YOLOv8m deep learning model to automatically detect weapons from live CCTV camera feeds. YOLOv8m is one of the most efficient object detection models available today. It can identify objects in images and videos quickly and accurately, making it a great choice for real-time surveillance applications.

Our model was trained on a labeled dataset of 16,000 images that include both guns and knives. After training and testing, our system achieved an accuracy of around 88% with a mAP50 of 88.54%. This shows that the model performs well and reliably predicts the presence of weapons in different conditions. The model does not just detect weapons in static images; it works on live video feeds in real time, which makes it practically useful.

One of the most important features of our system is the alert mechanism. Whenever a weapon is detected in the live CCTV feed, an instant alert message is displayed on the dashboard of our user interface. This means the security team or higher authorities do not have to keep watching the screen constantly; they will be notified the moment a threat appears. This fast and automated response can greatly help in preventing violent incidents in public spaces.

2. Literature Survey

Weapon detection in surveillance settings has become a popular research area because of the increasing demand for automated security systems. Researchers have tested various deep learning methods, from early CNN-based models to modern real-time object detectors like YOLO. This section reviews key existing studies and points out the gaps that the proposed system seeks to fill.

Khalid et al. (2023) introduced a weapon detection system for surveillance using YOLOv5 and Mask R- CNN for instance-level segmentation. They trained their system on a dataset of 2,971 images from the University of Granada, which they expanded to 7,131 images using data augmentation techniques such as flipping, rotation, blur, and cutout. The YOLOv5 model reached a precision of 94.63%, a recall of 96.25%, and an F1-score of 95.43%. Meanwhile, the Mask R-CNN achieved a detection accuracy of 90.66% and a mean Intersection over Union (mIoU) of 88.74%. Although the results were impressive, the dataset focused on just one type of weapon, and the system was not tested in real-world situations that involved both guns and knives.

Kambhatla and Ahmed (2022) created a firearm detection system using different YOLOv5 model variants trained on about 2,300 manually annotated images sourced from Kaggle and Google Open Images. The dataset included handguns, revolvers, rifles, knives, and people as detection categories. Their best model achieved a precision of 89.4%, a recall of 70.1%, and a mean Average Precision at 0.5 (mAP@0.5) of 80.5%, with the highest per-class precision reaching 94.1%. The paper noted that occlusion, size variation, and shape differences among firearm types remained major challenges for systems using a single model.

Yadav, Gupta, and Sharma (2024) proposed a real-time firearm detection system that combined YOLOv4 with a Super-Resolution CNN (SRCNN), calling it SRYolo. This model was designed to manage tough surveillance conditions, including blurred images, dark backgrounds, night scenes, and complex environments, which are often encountered in real-world CCTV footage. The model reached a precision of 85.22% and an F1-score of 80.98%. While the SRYolo method showed potential in low-quality image settings, its overall detection accuracy was lower compared to YOLOv5 and YOLOv8 variants. Additionally, it did not include a real-time alert system for security personnel.

Olmos et al. (2018) were among the first to use deep learning for automatic handgun detection in surveillance videos. They applied a Faster R-CNN model with VGG16 as the backbone and trained it on a dataset of 6,000 labeled images. The model achieved an accuracy of 91.43% but demonstrated a tendency to misidentify guns as visually similar objects like mobile phones, wallets, and cards, highlighting an ongoing issue with false positives that persists in later studies.

Hashmi et al. (2021) compared YOLOv3 and YOLOv4 on a custom dataset of 7,800 images collected from CCTV footage and Google Images, resizing all images to 448×448 pixels. YOLOv3 achieved an F1-score of 77%, while YOLOv4 improved this to 82%. This comparison illustrated the steady improvements in YOLO models over different versions, which provides a basis for using YOLOv8m in this study.

Bhatti et al. (2021) focused on weapon detection in real-time CCTV videos using deep learning and shared their findings in IEEE Access. Their study demonstrated that deep learning models can effectively operate for live video-based weapon detection but highlighted that many existing systems did not have a complete deployment pipeline that includes real-time alerts for security teams. This gap is directly addressed by our proposed system.

Overall, this review shows that while several systems have achieved good detection accuracy, most face one or more of the following issues: small or single-source datasets, no real-time alert system, lower performance in multi-class weapon detection (guns and knives together), and a lack of comparison among different YOLO versions on the same dataset. The proposed system tackles these challenges by training and comparing YOLOv5m and YOLOv8m on the same labeled dataset of 16,000 images, incorporating both guns and knives, and adding a real-time alert dashboard into the user interface.

2.1. COMPARSION TABLE

Authors & Year	Model Used	Dataset Size	Weapon Classes	Key Results
Khalid et al., 2023	YOLOv5 + Mask R-CNN	7,131 images (augmented)	Gun only	F1: 95.43% mloU: 88.74%
Kambhatla & Ahmed, 2022	YOLOv5 variants	~2,300 annotated images	Gun, Knife, Rifle	mAP@0.5: 80.5% Precision: 89.4%
Yadav et al., 2024	YOLOv4 + SRCNN (SRYolo)	Custom multi- condition data	Firearms	Precision: 85.22% F1: 80.98%
Olmos et al., 2018	Faster R-CNN (VGG16)	6,000 labeled images	Handgun	Accuracy: 91.43%
Hashmi et al., 2021	YOLOv3 / YOLOv4	7,800 images	Gun only	F1: 77% (v3) F1: 82% (v4)
Bhatti et al., 2021	CNN-based deep learning	CCTV footage	Gun	Real-time tested
Proposed Work	YOLOv5m + YOLOv8m	16,000 labeled images	Gun Knife	mAP50: 88.54% Precision: 87.54% (YOLOv8m)

3. Dataset Analysis

For this project, the dataset was sourced from Kaggle. The original dataset, Weapon and Knife Detection (137K Images) by zinkzsa, is one of the largest publicly available weapon detection datasets. It has about 137,000 images that cover two main weapon types: guns and knives. The images were collected from various sources, including internet images, CCTV footage frames, YouTube video captures, and real-life surveillance scenes. The dataset includes images taken under different conditions, such as indoor and outdoor settings, various lighting conditions, different camera angles, partial obstruction scenarios, and both close-up and distant views of weapons.

The second dataset, Weapons Dataset: Guns and Knives by manisaideepchintha, is a labeled dataset specifically prepared for object detection tasks. It has annotated images of guns and knives with YOLO- format bounding box labels. This format makes it directly usable with YOLOv5m and YOLOv8m training pipelines without needing extra annotation work.

3.1 Original Dataset vs. Modified Dataset Used in This Work

The original 137K dataset, while large, presented some practical challenges. Training a model on all 137,000 images needs high computational resources, a long training time, and strong GPU infrastructure that isn't always available in a standard research environment. Additionally, many images in the large dataset contain redundant or near-duplicate scenes that do not significantly contribute to model learning.

To solve these problems, we selected a subset of 16,000 images from the full 137K dataset. We chose these images to ensure maximum diversity, covering different weapon types, backgrounds, lighting conditions, and levels of obstruction while keeping the dataset size practical for training with available cloud GPU resources on Kaggle and Google Colab.

The key differences between the original dataset and the modified dataset used in this project are summarized below:

Property	**Original Dataset (137K)**	**Modified Dataset (Used in This Work)**
Total images	~137,000	16,000 (selected subset)
Classes	Gun, Knife	Gun, Knife
Annotation format	YOLO format labels	YOLO format labels (retained)
Training split	Not fixed	80% — 12,800 images
Validation split	Not fixed	20% — 3,200 images
Computational demand	Very high	Moderate — suitable for Kaggle/Colab GPU
Image diversity	Very high (redundancy present)	High (redundant images removed)
Training time	Very long	Practical and manageable

3.2 Dataset Split

After finalizing the 16,000 selected images, the dataset was divided into two parts using the standard machine learning practice of an 80/20 split:

- Training set — 12,800 images (80%): This set was used to train both YOLOv5m and YOLOv8m models. The training images encompass a wide range of real-world scenarios, allowing the model to learn general visual features of guns and knives.
- Validation set — 3,200 images (20%): This set was used to evaluate the model's performance after training.

The validation set was kept completely separate from the training data to ensure an unbiased assessment of detection accuracy, precision, recall, and mAP scores.

Both weapon classes, gun and knife, are included in the dataset with proper bounding box annotations in YOLO format. Each label file specifies the class index, center coordinates, and dimensions of the bounding box for each object in the image.

4. Methodology of Proposed System

The proposed smart surveillance system uses a deep learning-based approach for the detection of weapons in real-time surveillance videos. The main aim of the proposed methodology is to classify the input images or video frames as containing a weapon or not by learning discriminative features using the YOLOv8m object detection model.

Initially, a publicly available labeled data set was collected from Kaggle for training and evaluation purposes. The data set used for training and evaluation is Weapon and Knife Detection 137K Images, from which 16,000 images were carefully selected to ensure maximum diversity in different conditions. The images include gun and knife classes with appropriate bounding box annotations in YOLO format. Once the data set was finalized, it was split into 80% training data, which includes 12,800 images, and 20% validation data, which includes 3,200 images, to ensure appropriate learning and evaluation of the model. Before training the deep learning model, a preprocessing step was carried out, where images were resized to 640x640 pixels, normalized, and data augmentation techniques were applied to ensure that images are in a consistent format appropriate for training and testing with YOLOv8m model.

The proposed system is intended to automatically identify weapons from live CCTV camera feeds using YOLOv8m, which is one of the most advanced single-stage object detection models currently available. The system architecture is made up of a number of modules that are designed to work in tandem to process the input frames and produce output results. The input frame from the live CCTV camera feed is first processed by the image validation and preprocessing module. In this module, the system checks the image format, changes the size of the image to the required size of 640×640 pixels, and normalizes the pixel values to make it amenable to the YOLOv8m model. After the preprocessing, the image is then passed to the feature extraction and detection module. In this module, the YOLOv8m model analyzes the entire image in a single pass and also predicts the bounding boxes, class labels, and confidence scores of all detected objects in the image.

The YOLOv8m model generates a detection output that includes the coordinates of the bounding box, the predicted class label, which is either gun or knife, and a confidence score that represents the probability of the detection being correct. The outputs are then fed into the alert generation module. The alert module examines whether the confidence score of any detection exceeds the threshold value, and if so, the system immediately sends a real-time alert notification. Finally, the result visualization module shows the detection result to the security personnel via the Flask-based web interface dashboard. The system generates bounding boxes around the detected weapons on the live feed and shows an alert message along with the confidence score that represents the certainty of the detection. The proposed system has an overall detection accuracy of about 88%, which validates the effectiveness of the YOLOv8m model for real-time weapon detection in surveillance systems.

4.1 System architecture

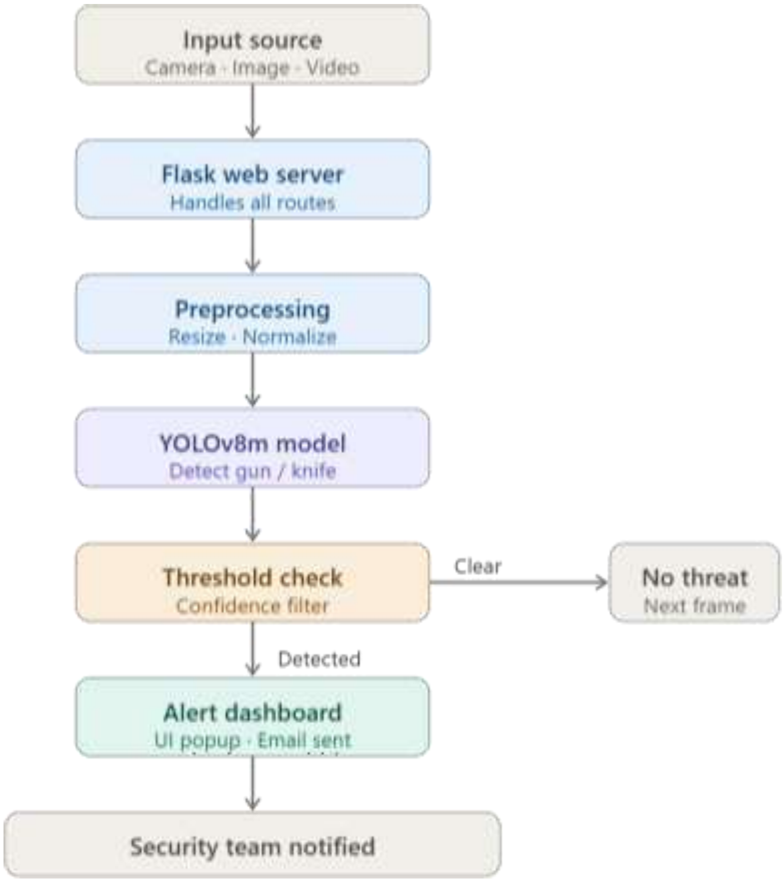

Fig:4.1.1 System architecture for smart surveillance system for firearm detection

The above figure shows the overall system architecture of the proposed Smart Surveillance System for real- time weapon detection. The input to the system can be taken from three different sources: a live CCTV camera feed, an uploaded image, or an uploaded video file, which is then routed through the Flask web server for processing. Each input is preprocessed and then routed to the YOLOv8m deep learning model for detecting the presence of weapons in the image. The final output is obtained from the confidence threshold check, which classifies the input as either weapon detected or clear, and the output is displayed on the alert dashboard with an alert notification sent to the security team.

The input source module receives three different types of inputs from the user. The first type of input is the live CCTV camera feed, which is processed frame by frame in real-time using the browser. The second type of input is the image file uploaded by the user, which can be in formats like JPG or PNG. The third type of input is the video file uploaded by the user, which can be in formats like MP4 or AVI. All three types of inputs are accepted using the same web interface, making the system versatile for use in different real-world surveillance scenarios.

The Flask web server is the central controller of the entire system. It is a lightweight web framework written in Python that handles all the routing and communication between the user interface and the backend processing modules. It receives the input from the user, handles authentication and sessions, routes the input to the appropriate processing pipeline, and then sends the annotated output back to the dashboard. All major operations of the system are controlled using the Flask server.

The preprocessing module is responsible for preprocessing each input before feeding it into the deep learning model.

All images or video frames are resized to a fixed size of 640x640 pixels. Normalization is also performed on each input. For live camera feed, Base64 decoding is performed on the browser frame to convert it into a NumPy array, which is compatible with OpenCV.

The YOLOv8m is the core object detection model used in this proposed system. "YOLOv8m" is short for "You Only Look Once version 8 medium." It is one of the most advanced one-stage object detection models in the computer vision domain. Unlike CNN, which is limited to image classification, or R-CNN and Faster R-CNN, which use a slow two-stage object detection process, YOLOv8m uses a fast one-stage object detection process by detecting objects in one pass. It uses a CSP backbone for deep feature extraction, a feature pyramid network for feature extraction at multiple scales, and an anchor-free object detection head for detecting objects. This is a custom-trained YOLOv8m object detection model, trained on 16,000 images of guns and knives.

The confidence threshold check filters the detection results obtained from the YOLOv8m model. This step ensures that only real detection results are used to generate alerts, while false detection results are filtered. For gun-related objects, the detection results must have at least a 20% confidence threshold. For knife- related objects, the detection results must have at least a 25% confidence threshold. When a detection result satisfies the threshold check, the system will trigger an alert. Otherwise, the frame will be marked as clean, and the system will proceed to the next frame without interruption.

The alert dashboard module is initiated when the weapon detection crosses the confidence threshold. Two things happen at this point. First, a popup alert message is sent in real-time to the Flask-based web dashboard, alerting the security personnel to the presence of a weapon. Second, an automatic email alert is sent to the user's registered email account via Gmail SMTP, which sends out the details of the weapon and the time at which the weapon was detected. There is a cooldown period of 30 seconds between emails to prevent repeated emails for the same ongoing weapon detection.

Lastly, the result of the detection is printed to the console, stating whether the weapon has been detected or not, and the type of weapon, whether it is a Gun or a Knife. All weapon detection events are recorded in the detection log, including the timestamp, username, file name, and detection details.

4.2 Block diagram

The block diagram of the proposed Smart Surveillance System for Weapon Detection shows the overall process from input to final output of the system. The inputs to the system are of three types: live CCTV camera feed, uploaded image, and uploaded video, which are sent to the flask web server after user authentication. These inputs are then sent to the preprocessing block, where the images are resized to 640x640 pixels and normalized, and then sent to the YOLOv8m deep learning model. The YOLOv8m model uses a CSP backbone, FPN neck, and anchor-free detection head to process the inputs and get the output from the deep learning model. The output is then sent to the confidence threshold block, where a detection of guns should have a confidence of at least 20%, and knives should have a confidence of at least 25%. If the weapon is detected, the alert block will immediately trigger a popup in real time on the dashboard, as well as send an email notification to the registered user via Gmail SMTP. Meanwhile, all the events of detection will be stored in the log block. Finally, the result will be displayed as either WEAPON DETECTED or NO THREAT, ensuring that the security team is immediately notified in the event that a dangerous weapon is detected in the video feed.

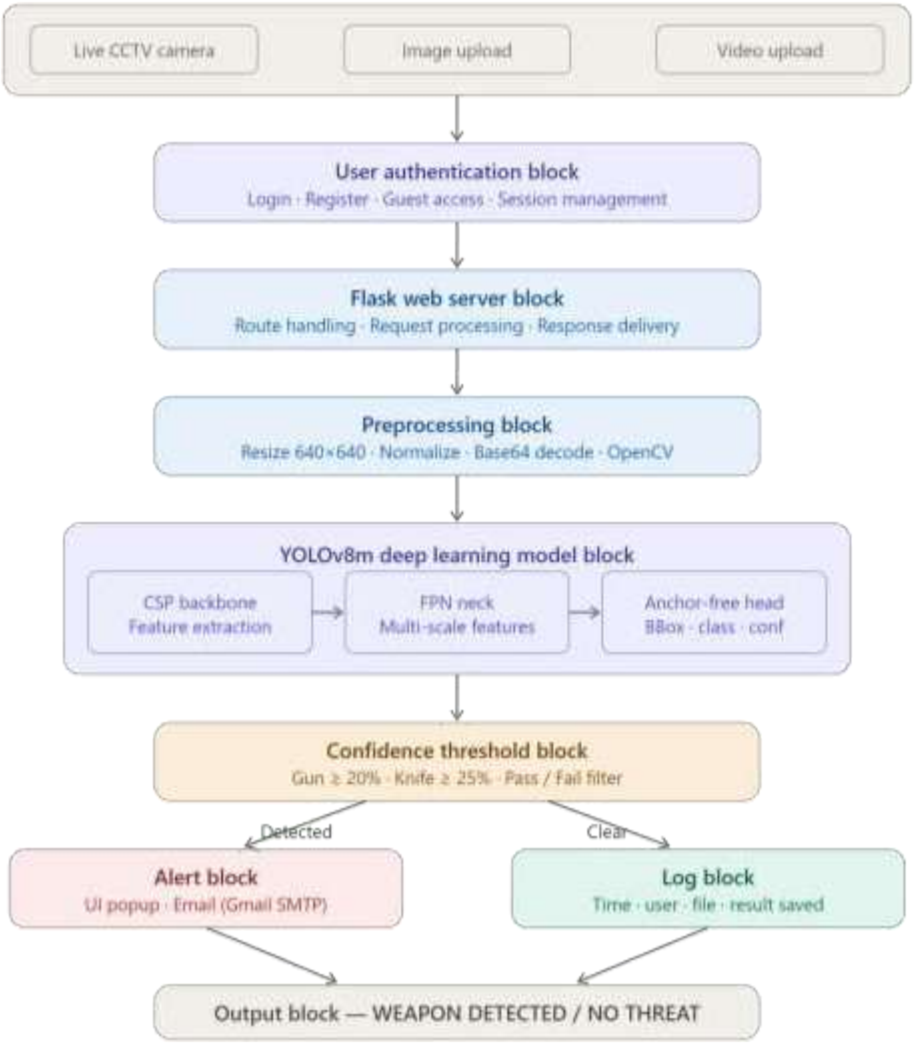

Fig:4.2.1 Block Diagram for Smart Survellianvce System for firearm Detection

5. Implementation

The smart surveillance system designed for the detection of weapons was implemented using a deep learning approach integrated with a web interface for real-time functionality. The system was implemented using the Python programming language and libraries such as Ultralytics, OpenCV, NumPy, Flask, and smtplib. The design and training of the system were done using cloud platforms like Google Colab and Kaggle, which support GPU for training and executing deep learning models. The model weights were saved on Google Drive and directly imported into the Flask application during execution, and the application was shared publicly using ngrok.

The dataset was collected based on the title and requirements of the project. The dataset was collected from Kaggle, keeping in mind the specific use case of detecting weapons in real-world surveillance systems. After the collection of the dataset, it was suitably modified to meet the requirements of the proposed model.
Initially, the dataset consisted of approximately 137,000 images. From this dataset, 16,000 images were used for training and validation. This was done to ensure that the dataset was balanced, diverse, and free from redundant or irrelevant images that could have negatively impacted the model. After the modification of the dataset, it was divided into training and validation sets consisting of 80% and 20%, respectively.

Before the training of the model, an image preprocessing pipeline was established. In this regard, all images were resized to have a resolution of 640 * 640 pixels, which is in line with the input requirements of the YOLOv8m model. Additionally, normalization and data augmentation processes, including flipping, rotation, and mosaics, were used to enhance the generality of the model. This is essential in ensuring that the model does not overfit or underfit the data, thus providing a well-balanced and reliable image detection model.

The Ultralytics library was used to import and utilize the YOLOv8m model for the weapon detection task. YOLOv8m was particularly selected for this project because it is the most appropriate model for this type of weapon detection task, as it is fast enough to be used in real-time surveillance systems and accurate enough to detect weapons clearly even in complex backgrounds. The medium version of the YOLOv8m model strikes the perfect balance between the size of the model and the accuracy of the detection, ensuring that the dataset does not result in overfitting due to the large size of the model or underfitting due to the small size of the model. The model was then fine-tuned using transfer learning on the collected and altered weapon dataset to allow it to learn the specific visual characteristics of guns and knives. A confidence threshold was also applied to ensure that only the most accurate detections are further processed, as gun-related detections are accepted only if the confidence score is above 20% and knife-related detections are accepted only if the confidence score is above 25%.

6. Experimental Results

6.1. Performance Comparison — Classification Report

The performance of the proposed system was tested on the validation dataset, which has 3,200 images. The YOLOv8m model was tested on two different classes of weapons, namely gun and knife, and compared with the results of the existing systems, as discussed in the literature survey. The metrics used for evaluating the performance of the system are Precision, Recall, mAP50, and mAP50-95, which are the most common metrics used for evaluating object detection systems.

As shown in the result image, the metrics of the proposed YOLOv8m model are as follows:

Metric	Value
Precision	87.54%
Recall	84.08%
mAP50	88.54%
mAP50-95	63.12%
Fitness Score	0.6312
Inference Speed	11.42 ms/image
Total Gun images	1461
Total Knife images	539

6.1.1. Comparison Table for Experimental Result

Authors & Year	Model Used	Dataset Size	Precision	Recall	mAP50
Khalid et al., 2023	YOLOv5 Mask R-CNN	7,131 images	94.63%	96.25%	95.43% (F1)
Kambhatla & Ahmed, 2022	YOLOv5 variants	~2,300 images	89.4%	70.1%	80.5%
Yadav et al., 2024	YOLOV4 SRCNN	Custom data	85.22%	—	F1:80.98%
Olmos et al., 2018	Faster R-CNN (VGG16)	6,000 images	-	—	91.43% (Acc)
Hashmi et al., 2021	YOLOv3 / YOLOv4	7,800 images	-	—	F1: 77% 82%
Bhatti et al., 2021	CNN-based DL	CCTV footage	-	-	Real-time only
Proposed Work (YOLOv8m)	YOLOv8m	16,000 images	87.54%	84.08%	88.54%

6.1.2. Confusion Matrix

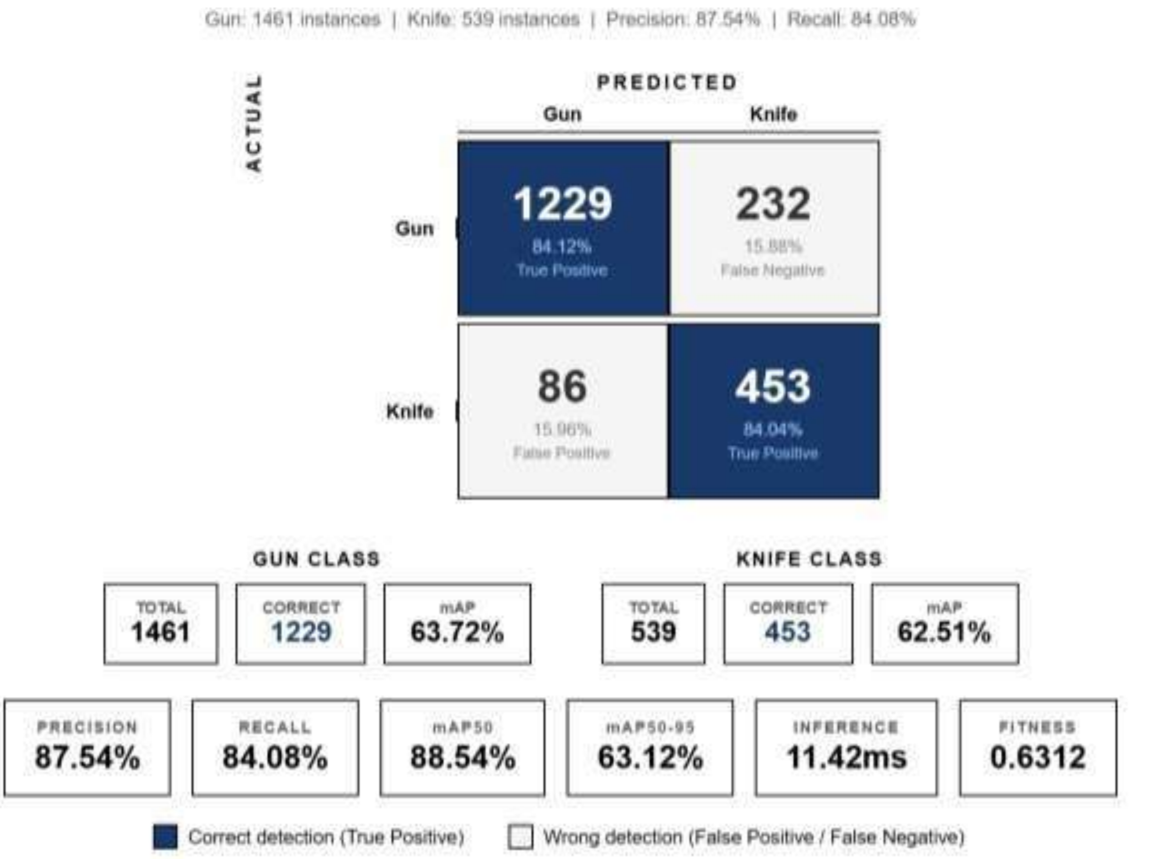

Fig:6.1.2. Confusion Matrix for the Smart Surveillance System For Firearm Detection.

6.2. Experimental Result of our project

The output result images of the proposed YOLOv8m weapon detection model are shown above, which illustrate the real-time detection ability of the system for various surveillance scenarios and types of weapons. In the first output image, the system has been able to detect two guns at the same time in a single image by placing two different red bounding boxes around the detected guns with confidence levels of 84.1% and 74.7%, respectively, which indicates that the system is able to detect multiple weapons at the same time in a single image, which is an essential requirement for real-time surveillance systems where more than one threat may be present. In the second output image, the system has been able to detect a rifle- type weapon from an actual CCTV footage image with a confidence level of 67.3%, which is above the defined threshold of 20% for gun-class detections, and the system has correctly drawn a red bounding box around the detected weapon with the class label clearly displayed on the frame, thus confirming that the model is working effectively even with poor-quality real-world CCTV images with different lighting conditions and image noises. In the third output image, the model has detected a knife being carried by a person in an outdoor street surveillance image with a confidence level of 32.9%.

fig:6.2.1. Gun detected using model yolov8m

fig:6.2.2. Gun detected using model yolov8m

fig:6.2.3.Knife detected using model yolov8m

7. Gaps Identified in the Proposed System

The proposed Smart Surveillance System, which utilized the YOLOv8m algorithm, was able to reach an accuracy level of 88% in terms of detection. In addition, the proposed Smart Surveillance System was also able to show satisfactory results in terms of reliable weapon detection. However, during the testing and evaluation phase of the proposed Smart Surveillance System, certain gaps were identified. Among the most significant gaps identified in the proposed Smart Surveillance System was the occurrence of false positives in the detection process. In this regard, it was identified that certain non-weapon objects were also detected by the proposed Smart Surveillance System as weapons. This is due to the fact that certain objects, which resemble guns or knives in terms of their shapes, sizes, or colors, were also detected by the proposed Smart Surveillance System as weapons.

The second significant gap that is observed is that, in some cases, the entire human body is detected along with the weapon as a combined entity instead of detecting and localizing just the weapon in the bounding box. In this case, the bounding box is drawn around the entire image of the person and the weapon together instead of being drawn tightly around just the weapon object. This is affecting the localization capability of this model. It is becoming difficult to accurately locate the weapon in the image. This is especially more difficult in scenarios where there are people present in the same scene at the same time. This is most probably due to the lack of diversity in the training data, where some images might have bounding boxes drawn around them including some context from the surrounding areas instead of just the region where the weapon is present. These are the two significant gaps that are observed in this system. These gaps are most probably due to the lack of diversity in the training data.

8. Conclusion

The sudden rise in violence involving weapons in public areas has created a pressing need for automated threat detection systems in contemporary security systems. Conventional surveillance systems, which rely entirely on human intervention, are prone to human limitations such as fatigue, slow response times, and the inability to simultaneously monitor multiple camera feeds. This project has attempted to overcome these drawbacks by designing a Smart Surveillance System for real-time weapon detection based on the YOLOv8m deep learning model, which is currently the most advanced single-stage object detection model.

The dataset for this project was sourced from Kaggle and was carefully curated to meet the specific requirements of the proposed system. Of the original 137,000 images in the dataset, 16,000 labeled images of guns and knives were chosen and divided into 80% training images and 20% validation images. The YOLOv8m model was then trained on the prepared dataset using transfer learning, with confidence levels of 20% for gun detection and 25% for knife detection to avoid false positives.

The experimental outcome showed that the proposed YOLOv8m model was able to provide a precision of 87.54%, recall of 84.08%, and mAP50 of 88.54% on the validation set. The model was able to successfully identify gun and knife type weapons under varied real-world surveillance scenarios such as indoor settings, outdoor street surveillance footage, and actual CCTV quality images. The inference time of 11.42 milliseconds per image further confirmed that the system is capable of processing live surveillance images in real time without any noticeable delay. A comparative analysis with existing solutions from the literature survey revealed that the proposed solution offers a balanced treatment of detection accuracy, multi-class detection capability, and real-time processing capability that most existing solutions lack.

Two primary gaps were identified during the analysis of the proposed system. The first gap is the existence of false positives, where the system occasionally tended to misclassify non-weapon objects as weapon objects in complex images. The second gap is that the system occasionally tended to plot bounding boxes encompassing the entire human body along with the weapon object instead of plotting the bounding box only around the weapon object, thus reducing the localization accuracy of the detection system.These issues are mainly due to the lack of annotation quality and diversity in the dataset, and they can be resolved in future research by utilizing more accurately annotated training data and a larger and more diverse dataset.

In conclusion, the proposed Smart Surveillance System has successfully proved that the deep learning-based weapon detection technique using YOLOv8m is a feasible, fast, and accurate method for the automation of threat detection in public surveillance systems. The proposed system is able to detect guns and knives from live CCTV streams, uploaded images, and video files, and it is a flexible and comprehensive solution for real-world security needs. Future improvements will aim at enhancing the accuracy of detection, minimizing false positives, expanding the system to support the detection of other categories of weapons, and integrating the model with edge devices for use in resource-limited scenarios.

9. References

[1] D. Suasnavas, D. Pachacama, and W. Oñate, "YOLO-V7 and YOLO-V8 Benchmark for Firearm Detection and Deep Learning Model Retraining," in Proc. CSEI 2023, Lecture Notes in Networks and Systems, vol. 775, Springer, Cham, 2024, pp. 167–181. https://doi.org/10.1007/978-3-031-69228-4_11

[2] P. Yadav, N. Gupta, and P. K. Sharma, "Real-Time Firearm Detection System Utilizing Deep Learning and Super-Resolution CNNs," in Proc. ICICC 2024, Lecture Notes in Networks and Systems, vol. 1021, Springer, Singapore, 2024, pp. 369–380. https://doi.org/10.1007/978-981-97-3591-4_30

[3] A. Kambhatla and K. R. Ahmed, "Firearm Detection Using Deep Learning," in Intelligent Systems and Applications, IntelliSys 2022, Lecture Notes in Networks and Systems, vol. 544, Springer, Cham, 2023, pp. 200–218. https://doi.org/10.1007/978-3-031-16075-2_13

[4] S. Khalid, A. Waqar, H. U. A. Tahir, O. C. Edo, and I. T. Tenebe, "Weapon Detection System for Surveillance and Security," in Proc. IEEE ITIKD 2023, IEEE, 2023, pp. 1–6. https://doi.org/10.1109/ITIKD56332.2023.10099733

[5] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using

Deep Learning," IEEE Access, vol. 9, pp. 34366–34382, 2021. https://doi.org/10.1109/ACCESS.2021.3059170

[6] T. S. S. Hashmi, N. U. Haq, M. M. Fraz, and M. Shahzad, "Application of Deep Learning for Weapons Detection in Surveillance Videos," in Proc. IEEE ICoDT2 2021, IEEE, 2021, pp. 1–6. https://doi.org/10.1109/ICoDT252489.2021.9587423

[7] R. Olmos, S. Tabik, and F. Herrera, "Automatic Handgun Detection Alarm in Videos Using Deep Learning," Neurocomputing, vol. 275, pp. 66–72, 2018. https://doi.org/10.1016/j.neucom.2017.05.012

[8] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond," arXiv:2304.00501, 2023. https://arxiv.org/abs/2304.00501

[9] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," arXiv:2305.09972, 2023. https://arxiv.org/abs/2305.09972

[10] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in Proc. IEEE/CVF CVPR, 2023, pp. 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00736

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017. https://doi.org/10.1109/TPAMI.2016.2577031

[12] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications," Multimed. Tools Appl., vol. 82, no. 6, pp. 9243–9275, 2023. https://doi.org/10.1007/s11042-022-13644-y

[13] P. Yadav, N. Gupta, and P. K. Sharma, "Robust Weapon Detection in Dark Environments Using YOLOv7-DarkVision," Digit. Signal Process., vol. 145, 2024. https://doi.org/10.1007/s41870-024-01889-9

[14] A. Warsi, M. Abdullah, M. N. Husen, and M. Yahya, "Gun Detection System Using YOLOv3," in Proc. IEEE ICSIMA, 2019, pp. 1–4. https://doi.org/10.1109/ICSIMA47Sync.2019.9254360

[15] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Proc. ECCV 2016, Lecture Notes in Computer Science, vol. 9905, Springer, Cham, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[16] G. Jocher, "Ultralytics YOLOv8," GitHub Repository, 2023. Available: https://github.com/ultralytics/ultralytics

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE CVPR, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[18] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. ICML, 2019, pp. 6105–6114. https://arxiv.org/abs/1905.11946

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020. https://arxiv.org/abs/2004.10934

[20]. D. Qi, W. Tan, Z. Liu, Q. Yao, and J. Liu, "A Dataset and System for Real-Time Gun Detection in Surveillance Video Using Deep Learning," 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC).
Link: https://arxiv.org/abs/2105.01058

[21]. M. T. Bhatti et al., "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," IEEE Access, vol. 9, 2021.
Link: https://ieeexplore.ieee.org/

[22]. T. Murugan et al., "AI-Based Weapon Detection for Security Surveillance: Recent Research Advances (2016–2025)," Electronics (MDPI), 2025.
Link: https://www.mdpi.com/2079-9292/14/23/4609

[23]. S. Yellapragada et al., "CCTV-Gun: Benchmarking Handgun Detection in CCTV Images," arXiv:2303.10703, 2023.
Link: https://arxiv.org/abs/2303.10703

[24]. A. Thakur et al., "Real-Time Weapon Detection Using YOLOv8 for Enhanced Safety," arXiv:2410.19862, 2024.Link: https://arxiv.org/abs/2410.19862

[25]. A. Jadhav et al., "Confidence Aware SSD Ensemble with Weighted Boxes Fusion for Weapon Detection," arXiv:2509.23697, 2025.Link: https://arxiv.org/abs/2509.23697