International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Intelligent Violence Detection: An AI Driven Approach for Enhanced Surveillance System

Ahilya Sarnaik[1], Ms. Komal Jadhav[2], Rohan Sardesai[3], Sarvesh Padwal[4], Akhilesh Huddar[5]

[1,2,3,4,5]KIT's College of Engineering (Autonomous), Kolhapur, Maharashtra, India

Abstract

Violence Alert System is an AI surveillance system based on machine vision designed to detect violent attacks and threats. Advanced machine learning is implemented in the system to boost public safety by enabling human detection, violence identification, and the detection of new weapons. Real-time detection of humans uses the Faster R-CNN structure, and detection of violent activity uses MobileNetV2. Incorporation of weapon detection improves threat detection by identifying firearms and knives within videos. The system provides instan- taneous notifications using Telegram, whose images are location- time-stamped, hence supporting on-time response from law enforcement bodies. Detection enhancement of faces and images using MTCNN provides accurate recognition, with Firebase Cloud Firestore allowing safe data retrieval and storage. With multilayered threat assessment being used, the Violence Alert System provides an efficient and scalable solution for active public security monitoring in business sectors, schools, and public areas.

Keywords: Violence Detection, Real-time Surveillance, Weapon Detection, Faster R-CNN, AI-powered Security System.

INTRODUCTION

Violent public incidents have been more frequent in recent years hence, there is a pressing need for pre-emptive safety measures. Traditional surveillance equipment, despite being widely used, is intelligence-hungry and is unlikely to identify potential threats on its own, relying solely on visual inspec- tion and response time. Violence Alert System is therefore proposed as an AI-facilitated real-time surveillance system that can identify violent behavior, weapon-carrying, and suspicious human presence in public areas.

For multilayered threat prediction, the architecture uses the most advanced deep learning models. Real-time people detection is delivered through the use of Faster R-CNN ar- chitecture, enabling effective detection of humans in video streams. MobileNetV2 is used in the detection of violent behavior, as it is computationally cost-efficient and efficient on embedded systems. The system also includes mechanisms for weapons detection that can identify firearms, as well as bladed weapons such as knives.

One of the most important features of the system is that it transmits time-stamped images and geolocation data to author- ities instantly in real-time through Telegram to enable instant response. Face and image recognition are enhanced by using the MTCNN (Multi-task Cascaded Convolutional

Networks) to improve the accuracy of individual identification when the surroundings pose challenges. Firebase Cloud Firestore is used for event details and alarm history records to provide reliability as well as scalability in secure data storage.

By combining advanced AI models with trustworthy data management and communication solutions, the suggested Vi- olence Alert System gives a brilliant, scalable, and efficient real-time public safety watch solution. The potential scope for its application includes schools, office spaces, public transport stations, and other vulnerable environments, indicating that it represents a major technological innovation in security infrastructure automation.

LITERATURE REVIEW

The recent developments in computer vision and deep learning have played an important role in assisting in the devel- opment of smart surveillance systems that can detect threats in real-time. Architectures such as faster R-CNN have been able to. Detect humans and objects quite well because they are both. fast and accurate when processing video frames. Mo- bileNetV2, being lightweight, has increasingly been employed for behavior analysis applications like violence detection and is therefore optimally suited for real-time deployment in low- resource settings. Weapon detection has also been enhanced through CNN-based detectors like YOLO and SSD, which can detect guns and knives with decent accuracy in occluded visual environments.

TABLE I LITERATURE REVIEW

Year	Author	Title	Methodology
2019	M. Ramzan et al.	A Review on State-of-the-Art Violence Detec- tion Techniques [1]	The paper surveys violence detection methods using traditional ML, SVM, and deep learning techniques, highlighting models with reported accuracies up to 97%.
2021	Sarthak Sharma, B Sudharsan, Saamaja Nara- harisetti, Vimarsh Trehan, Kayalvizhi Jayavel	A fully integrated violence detection system using CNN and LSTM [3].	The paper presents a real-time system for detecting violence using CNN and LSTM that is capable of alerting the authorities in real time via a mobile application. Since the au- thors attained low performance on unrealistic benchmark data, they used a modified UCF Crime dataset of 160 trimmed videos. With over 50 epochs of training with an 80/20 split, the model attained a very high testing accuracy of 98.87%.
2024	Shahriar Jahan, Roknuzzaman,	Deep Learning-Based Human	The paper emphasizes the

	Md Robiul Is- lam	Detection for Smart Surveillance Systems [5]	accuracy of dif- ferent machine learning and deep learning methods for video-based Human Activity Recognition (HAR) in surveillance systems. Deep learning models, specifically CNNs and RNNs, show accuracy rates between 85% and 95%, which is superior to conventional ap- proaches such as HMM and K-means clus- tering. The paper stresses the necessity of real-time, intelligent surveillance to identify unusual activities in public places with heavy crowds.
2019	Chloe Eunhyang Kim, Mahdi Maktab Dar Oghaz, Jiri Fajtl, Vasileios Argyriou, Paolo Remagnino	A Comparison of Embedded Deep Learning Methods for Person Detection [6]	The paper ”A Comparison of Embedded Deep Learning Methods for Person Detection” eval- uates multiple object detection models on em- bedded platforms like NVIDIA Jetson TX2. It compares models such as YOLOv3-416, Tiny YOLO, SSD, Faster R-CNN, and R-FCN using an indoor dataset of over 10,000 images. YOLOv3-416 achieved a balanced performance with around 61.1% mAP (mean Average Precision) while maintaining real- time speed, making it ideal for embedded use. In contrast, Faster R-CNN with Inception ResNet-v2 delivered the highest accuracy of 76.4% mAP, though with slower inference times. Tiny YOLO and SSD were the fastest but had lower accuracy, around 33–45% mAP. The study concludes that

			YOLOv3-416 offers the best trade-off between speed and accuracy for real-time person detection.
2021	Muhammad Tahir Bhatti, Muhammad Gufran Khan, Masood Aslam, and Muhammad Junaid Fiaz	A Comparison of Embedded Deep Learning Methods for Person Detection	The paper ”A Comparison of Embedded Deep Learning Methods for Person Detection” com- pares various object detection models on em- bedded hardware such as NVIDIA Jetson TX2. It compares models such as YOLOv3- 416, Tiny YOLO, SSD, Faster R-CNN, and R-FCN using an indoor dataset of more than 10,000 images. YOLOv3-416 provided a bal- anced performance with approximately 61.1% mAP (mean Average Precision) and achieved real-time speed, thus being perfect for em- bedded applications. Conversely, Faster R- CNN with Inception ResNet-v2 produced the highest accuracy of 76.4% mAP, albeit with reduced inference times. Tiny YOLO and SSD were the quickest but less accurate, at approxi- mately 33–45% mAP. The research concludes that YOLOv3-416 provides the optimal bal- ance between speed and accuracy for real-time human detection.
Year	**Author**	**Title**	**Methodology**
2023	Rajdeep Chatterjee, Manas Ranjan Pradhan, Biswaranjan Acharya, Ankita Chatterjee And Tanupriya Choudhury.	A Deep Learning-Based Efficient Firearms Monitoring Technique for Building Secure Smart Cities [9]	The paper presents a deep learning-based system for detecting firearms and human faces to address firearm-related violence. Us- ing models like Faster R-CNN and Efficient-Det, along with ensemble

			methods such as Weighted Box Fusion, the system enhances detection accuracy. It achieves mAP scores of 77.02% (@0.5), 16.40% (@0.75), and 29.73% (@0.5:0.95). This approach improves surveil- lance efficiency and can also help detect gun- related content in social media videos.
2021	Paolo Sernani, Nicola Falcionelli, Selene Tomassini, Paolo Contardo, and Aldo Franco Dragoni	Deep Learning for Automatic Violence Detec- tion: Tests on the AIRTLab Dataset [10]	The proposed deep learning models achieved an overall accuracy of approximately 88% on the AIRTLab dataset, highlighting their effectiveness in detecting violent actions while maintaining robustness against false positives caused by friendly interactions such as hugs, claps, or high-fives. This confirms the relia- bility of transfer learning-based 3D CNNs for spatio-temporal video analysis.
2024	MUSTAQEEM KHAN WAIL GUEAIEB AB-DULMOTALEB EL SADDIK, GIULIA DE MASI AND FAKHRI KARRAY [11]	VD-Net: An Edge Vision-Based Surveillance System for Violence Detection	This work introduces VD-Net, a lightweight AI-based violence detection framework de- signed for real-time surveillance using Intel- ligent IoT (IIoT). It leverages ST-TCN blocks and bottleneck layers to extract key temporal features and classify violent vs. non-violent actions. The system can also trigger alerts when violence is detected. Both surveillance and non-surveillance dataset evaluations reveal a 1–4% accuracy gain over cutting-edge mod- els, demonstrating its dependability and effec- tiveness for low-power real-time applications.

METHODOLOGY

A. Human Detection using RCCN

Dataset Collection and Preprocessing: We used the Person Detection dataset from Kaggle, comprising a wide range of high-resolution images appropriate for object detec- tion purposes. The data features annotated bounding boxes for people across different real-world scenes. In training using an R-CNN model, images were preprocessed by rescaling to a standard resolution of 224×224 (in keeping with the majority of CNN backbones such as ResNet). Other steps involved nor- malization and data augmentation (random cropping, flipping, brightness changes) to improve generalization and increase model robustness to occlusions and different poses.

Fig. 1. Sample set of pre-processed training images.

Structural Layout of the Region-based Convolutional Neural Network (RCNN): As a human detector, the sys- tem employed a Region-based Convolutional Neural Network (RCNN) in the Faster R-CNN architecture with a ResNet-50 backbone merged with a Feature Pyramid Network (FPN). The model was pre-trained on the COCO dataset and then fine- tuned on a Kaggle human detection dataset. The architecture components are explained as follows:

- Backbone Network (ResNet-50 with FPN): takes the in- put image and extracts extensive hierarchical features. It generates feature maps with multiple scales.
- Region Proposal Network (RPN): Slides over the feature maps to produce object proposals (regions probably hav- ing an object).
- RoI Align Layer: Extract fixed-size feature maps from proposals through bilinear interpolation to prevent mis- alignments.

Detection Head:

- Two Fully Connected (FC) layers for object classifi- cation and bounding box regression.
- Employs Softmax for object classification (here, 'person' and 'background').
- Bounding Box Regressor: Adjusts the proposed boxes to better locate the detected person.
- Training Loss: Sum of classification loss and bounding box regression loss. Optimizer used is Stochastic Gradient

Descent (SGD) with momentum. The final model was trained for 3 epochs on person images that are annotated, and saved as

Choice of RCNN Model: Based on accuracy and de- ployment in our person detection pipeline, we chose the Region-based Convolutional Neural Network (RCNN), or in this instance Faster R-CNN with ResNet-50 backbone and Feature Pyramid Network (FPN).

RCNN-based models are typically characterized by their object detection accuracy, which was paramount in detecting humans in cluttered environments. Faster R-CNN has a Re- gion Proposal Network (RPN) that generates object proposals rapidly and is therefore much faster than earlier versions of RCNN while maintaining state-of-the-art performance.

The ResNet-50 backbone was chosen because of its residual learning technique, which is said to be able to extract deep hierarchical features without having a tendency to disappear through disappearing gradients. Coupled with an FPN, the resulting architecture enables consistent detection at diverse scales—a characteristic of important value considering human variability in terms of sizes and orientations in images.

In comparison with one-stage detectors like YOLO or SSD, Faster R-CNN can yield higher detection accuracy, and ours was well more aligned towards the safety-critical nature of our violence and weapons detection system. While for a cost, an additional computational cost, the compromise proved to be necessary because it elevated detection dependability. That pre-trained weights existed and that transfer learning is enabled also helped reduce training time while achieving good performance with an even moderately sized dataset.

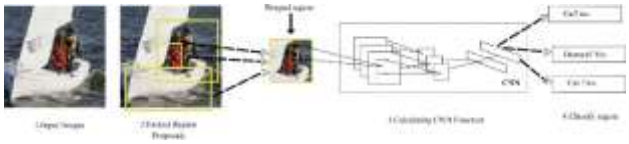

Fig. 2. Architecture of RCNN model

The RCNN algorithm, particularly Faster R-CNN, also contributed to the interpretability and trust in the results obtained. Although more advanced than a basic CNN, the modular design of Faster R-CNN—distinct stages for feature extraction, region proposal, and classification—allowed for a greater insight into how the model arrived at its detection decisions. Such interpretability is valuable in safety-related applications such as violence and weapon detection, where system transparency and trust are critical.

Besides, a review of past research and detection thresholds in comparable tasks shows that Faster R-CNN performs better than most alternatives across the board, particularly in preci- sion and accuracy. The popularity of the algorithm in human and object detection validated its suitability and justified its selection for our project.

In a nutshell, the Faster R-CNN model was used with ResNet-50 and FPN due to its shown detection performance, consistent interpretability of data, and track record in related tasks. These characteristics are particularly suitable to the goals of our project, where reliable and accurate human detection is a primary concern.

Transfer Learning and Training: We let the model start learning based on features that were previously found suitable for the detection of generic image attributes. After the process of learning had been finished, fine-tuning was employed to shape the features to align with the features of the medicinal plant database. The training process comprised steps like partitioning the dataset into 70- 30% wherein training data receives 70 percent of the dataset and the other 30 percent was utilized as test data. We applied the concept of early stopping on the validation loss so that we wouldn't overfit and get relatively good model outputs. For the 20-epoch training phase, the model was trained on a batch size of 32 to have an acceptable balance between computational efficiency and model convergence.

B. Violence Detection using MobileNetV2

Dataset Collection and Preprocessing: The dataset used in this project is the RWF-2000 (Real World Fights) dataset. The dataset contains 2,000 video clips, each divided uniformly into two classes: **Fight** and **NonFight**. The videos are surveil- lance videos of real world, and hence, the dataset is ideal for being used in real-world public surveillance and monitoring systems. For pretraining the dataset, the following are the steps to be used:

Fig. 3. Sample annotated images from the Violence dataset showing violence frame extracted from avi video.

Frame Extraction: An equal number of frames (usually 30) are sampled from all videos to sample the temporal aspects of actions.
Resizing: To preserve the input size required by Mo- bileNetV2, each frame is shrunk to a constant size of 224x224 pixels.
Normalization: For increased training stability and ef- fectiveness, pixel values are normalized between [0, 1].
Data Splitting: To enable appropriate evaluation, the data is divided into training, validation, and test sets.
Batch Generator: An in-house built data generator is used to load training batches of frames from videos without occupying the entire system memory.
Structural Model Layout of MobileNetV2: The architec- ture is a two-part hybrid model: spatial feature extraction using CNN and temporal modeling using LSTM.
Spatial Feature Extraction: Each frame of a video is passed through a pre-trained MobileNetV2 network using a TimeDistributed wrapper. This allows the model to extract features from each frame independently.
Temporal Modeling: The LSTM layer is then input with the sequence of frame features that have been extracted. It captures the temporal relations between frames.
Classification Layer: The LSTM output is passed through a single or several dense layers, the final output layer using a sigmoid activation to predict whether the video into violent or not.
This configuration enables the model to comprehend the video sequence's visual content as well as its movement patterns.
Model Choice: MobileNetV2 + LSTM is used as a com- bination model because it achieves the best tradeoff between performance and efficiency.

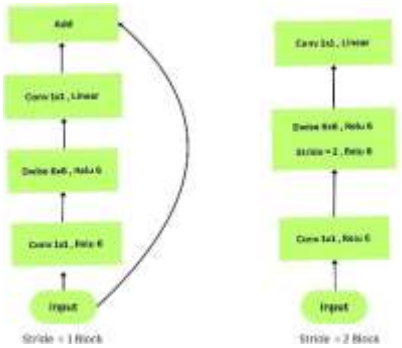

Fig. 4. MobileNetV2 architecture

MobileNetV2: A light, efficient CNN architecture opti- mized for mobile and embedded vision applications. It learns high-level spatial features at fewer parameters.

LSTM: Best suited to learn long-term dependencies in sequence data, allowing the model to realize how actions unfold over time.

Advantages: This configuration allows the model to analyze spatial and temporal features at virtually no computational cost, which is appropriate for real-time applications.

Transfer Learning and Training: For faster learning and better accuracy, transfer learning is utilized with a pre-trained MobileNetV2 model:

Initial Training: The convolutional base in MobileNetV2 is kept frozen to retain pre-learned features learned from ImageNet. Only the LSTM and dense layers are trained first.

Fine-Tuning: Towards the later part of the training phase, certain layers of MobileNetV2 are unfrozen and trained with a decreased learning rate in order to adjust the model towards the task of violence detection.

Loss and Optimization: The Adam optimizer is chosen because of its adaptive learning properties, and binary cross-entropy is used as the loss function.

Callbacks: Methods such as EarlyStopping and Mod- elCheckpoint are employed to avoid overfitting, and the best-performing model is retained.

This step-wise training approach benefits from exploiting earlier knowledge, yet nevertheless permits the model to accommodate newer, domain-dependent patterns.

C. Weapon Detection using YOLOv8

Dataset Collection and Preprocessing: The weapon de- tection component of this project utilized the *Gun3* dataset hosted on Roboflow[1].More than 1,000 annotated photos of various weaponry, including as pistols, revolvers, and rifles, taken in various real-world situations make up this dataset. The scenarios include a variety of lighting conditions, angles, and occlusions, ranging from controlled inside circumstances to complex, crowded public spaces. This dataset's diversity contributes to the model's strong generalization across many real-world contexts.

To ensure consistency and compatibility with the YOLOv8 architecture, all images were resized to 640×640 pixels. The preprocessing tools in Roboflow were used to transform label annotations from their original XML or Pascal VOC format into COCO JSON format. This conversion includes class labels and bounding box coordinates, which were normalized for efficient learning. Additionally, the dataset is split into training, validation, and test sets, with 80

Several data augmentation strategies were employed during training to enhance the model's generalization ability:

Mosaic Augmentation: Combines four random images into one, introducing multiple objects at different scales. This improves the model's capacity to manage intricate scenarios with several guns active at once.

CutMix Augmentation: Merges two images by over- laying one onto the other, enhancing robustness against occlusions, which are frequent in real-world scenarios like crowded areas or partial object visibility.

Color Jittering and Random Brightness: Simulates variable lighting environments, which is crucial for real- time applications where lighting conditions may change unpredictably.

Horizontal Flip and Random Rotation: Augments geometric diversity, helping the model detect firearms from different angles and orientations.

These augmentations significantly increase dataset variabil- ity, reducing overfitting and improving real-world detection performance.

[1]https://universe.roboflow.com/test11-iml4z/gun3-rove6/dataset/2

Fig. 5. Sample annotated image from the Roboflow Gun3 dataset showing bounding boxes and firearm labels.

Model Architecture and Components: YOLOv8, the latest object detection framework released by Ultralytics, was selected for this project due to its speed, accuracy, and modern features such as anchor-free detection. It is implemented in PyTorch with a modular architecture and integrates many en-hancements over earlier YOLO versions. YOLOv8's modular design allows for easy experimentation and fine-tuning based on specific requirements.

Backbone (CSPDarknet): Utilizes Cross Stage Partial connections to retain gradient flow and enable deeper networks with fewer parameters. It extracts high-level spatial features efficiently, which are crucial for detecting small and large objects in complex environments.

Neck (PANet - Path Aggregation Network): Facilitates multi-scale feature fusion, which is crucial for detecting objects at different scales. This allows YOLOv8 to handle both small, distant objects and

larger, closer objects with ease.

Detection Head: Uses anchor-free object detection, which directly predicts box centers and sizes. This ap- proach eliminates the need for predefined anchor boxes, simplifying the model and improving inference speed and accuracy.

Loss Function: Combines CIoU (Complete Intersection- over-Union) loss for accurate localization with Binary Cross-Entropy for class and objectness prediction. This multi-part loss function optimizes both the location and class of detected objects.

Optimizer and Scheduler: SGD with momentum was used, along with cosine annealing for learning rate decay, ensuring stable convergence while avoiding overfitting.

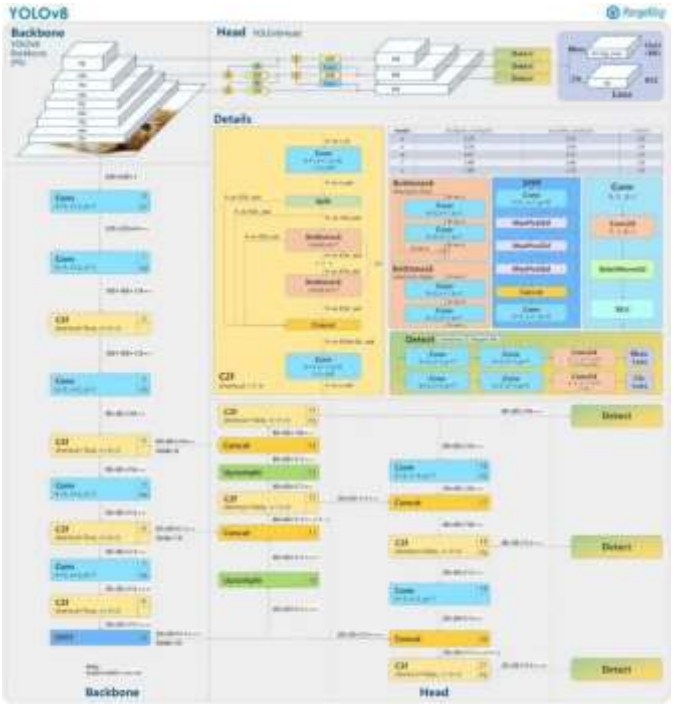

Fig. 6. YOLOV8 architecture structure as illustrated by Viso.ai [25].

Training Procedure and Evaluation Metrics: The model was trained using the official Ultralytics implementation of YOLOv8. The following parameters were configured for op- timal performance:

- **Epochs**: 100
- **Batch Size**: 16
- **Image Resolution**: 640 × 640 pixels
- **Learning Rate**: Cosine scheduler with initial LR = 0.01
- **Validation Split**: 20% of the dataset held out for valida- tion
- **Hardware**: NVIDIA RTX 3060 GPU with 12GB VRAM Model performance was evaluated using:
- **Precision and Recall**: Precision evaluates how well the model can detect firearms, or true positives, while lowering false positives, or incorrect detections. Recall reduces false negatives by assessing the model's ability to detect all relevant cases (firearms).
- **Mean Average Precision (mAP@0.5 and mAP@0.5:0.95)**: This statistic provides a more thorough assessment of model performance in a range of localization circumstances by comparing detection accuracy across different Intersection over Union (IoU) thresholds.
- **Inference Speed (FPS)**: This is a critical performance measure, especially for real-time applications.

The model was evaluated based on its ability to process frames at 30+ FPS on consumer-grade GPUs and even on edge devices.

The final trained model achieved:

- **Precision**: 91.2%
- **Recall**: 89.7%
- **mAP@0.5**: 88.5%
- **Inference Speed**: 32 FPS (real-time capable)

Deployment and Real-Time Inference: To facilitate real- world deployment, the trained model was converted to ONNX format, ensuring compatibility with edge computing devices such as NVIDIA Jetson Nano, Jetson Xavier NX, and Rasp- berry Pi (with Coral TPU acceleration). The ONNX for- mat enables easy deployment across multiple platforms and ensures fast inference times, even on hardware with lower computational power.

The deployed inference pipeline performs the following steps:

1. Capture frames from a video feed or camera in real-time.
2. Preprocess frames (resize, normalize) and pass them to the YOLOv8 model for prediction.
3. Detect firearms and draw bounding boxes around the objects with class and confidence scores.
4. Generate logs or real-time alerts when a firearm is detected above a pre-defined confidence threshold, trig- gering security protocols.

The system can process 30+ FPS on a consumer-grade GPU, with optimized edge devices (such as Jetson or Raspberry Pi) capable of maintaining low latency, making the solution practical for real-time surveillance.

Use Case Integration: This weapon detection module can be integrated into a multi-modal safety analytics system, including:

School and Campus Security: Automated alerts during unauthorized weapon detection, improving the response time of security personnel and reducing the risk of violence.

Public Transportation and Airports: Screening of surveillance feeds for concealed firearms, helping ensure public safety without requiring invasive security mea- sures.

Smart Cities and Law Enforcement: Integrated with existing CCTV infrastructure for continuous monitoring, assisting law enforcement agencies in identifying threats and responding promptly.

Real-Time Dashboarding: Combined with dashboards for live detection and log visualization, offering security teams a comprehensive view of all active surveillance feeds.

In future developments, the model may be extended to include:

- Multi-class weapon detection (e.g., knives, explosive de- vices) to extend the system's capability for detecting a wider range of potential threats.
- Human-weapon interaction detection, enabling the iden- tification of threatening situations such as armed robbery or attacks.
- Event classification (e.g., armed robbery, brandishing, shooting), which would help classify detected events and trigger appropriate responses based on the context.
- Integration with violence recognition modules for a more comprehensive security system that can recognize violent events in conjunction with weapon detection.

This makes the YOLOv8-based weapon detection system not only fast and accurate but also scalable and adaptable for various real-world security applications.

D. Telegram Alerts

A Telegram bot was added to the pipeline to generate real-time notifications, improving the utility of the violence detection system and weapon in day-to-day situations. The capability offers remote surveillance and quick response, es- pecially in dangerous settings like workplaces, schools, train stations, and other secure areas. With auto-alerts to concerned authorities or users whenever suspicion arises, the system improves security and responsiveness.

Telegram Bot Integration: A bespoke Telegram bot was developed with the help of BotFather, the native Telegram bot administration platform. The bot was then integrated into the detection pipeline through the Telegram Bot API so that end-user interaction with the detection model is streamlined. Key Features:

- Configurable Chat IDs: Alerts will be sent to a single user, Telegram group, or security channel, depending on the dynamic chat ID configuration.
- Lightweight API Calls: The integration is based on lightweight HTTP calls using libraries like requests or python-telegram-bot with minimal overhead and optimized message delivery.
- **Multimedia Support:** The bot can deliver text notifica- tions, along with annotated images (images of detections with bounding boxes and labels), providing more context per notification.
- **Instant Notifications:** The notifications are delivered in seconds of detection, which is appropriate for safety- critical applications.

Threshold-Based Alert Triggering: To restrict false alarms and maintain reliability, a threshold-based alert mecha- nism is used. The model tracks the number of frames detected as ”Violent” within a sliding window of current frames (e.g., 30 seconds or past 100 frames).

- **Alert Criteria:** An alert is triggered whenever the pro- portion of violent frames exceeds an initially set value of 25%, i.e., Violent Frames/ Total Frames ¿ 0.25
- **Snapshot Attachment:** It has an attached sample frame from the video, with bounding boxes, class labels (e.g., ”Violence”), and related confidence scores.
- **Metadata Inclusion:** The notification message also con- tains the detection time and, if known, contextual data such as GPS position or zone of the region under surveil- lance (e.g., ”East Wing Camera”).

This technique guarantees the reporting of only redundant and significant events and prevents the triggering of false alarms from transient motion or noise.

Alert Sample Format: The default alert message format reported by the Telegram bot is as follows:

- **Alert:** Violence Detected!
- **Confidence:** 87
- **Snapshot:** [Attached highlighted image]
- **Time:** 14:32:15
- **Location:** East Wing Surveillance Camera

This compact form displays information needed at a glance and may optionally include an instant link to view the video clip or store the recording for review and storage. Situational awareness by the recipient is eased to a great extent through the use of a marked photograph.

System Benefits:

- **Less Manual Monitoring:** Evades the need for con- tinuous manual monitoring by performing detection and alerting independently.

- **Real-Time Alerting:** Real-time alerting enables real-time action or investigation.
- **Runs on Edge Devices:** It may be run on low-resource devices with little access to the internet, and so it can find application in use in low-resource or rural communities.
- **Scalability:** The system can be scaled to multiple surveil- lance regions by the distribution of individual chat IDs or the distribution of individual Telegram groups for a region or camera.

Generally speaking, the Telegram alert integration enables the detection system to become more than just a passive surveillance mechanism and is instead a proactive and re- active protective mechanism, optimally applicable in real- time countermeasures for threats across various operational environments.

RESULTS AND ANALYSIS

All of the suggested Intelligent Violence Detection System's components were tested separately to confirm its functionality. According to the results, the integrated AI models are effective and process information in real time.

A. Human Detection

The Feature Pyramid Network (FPN) and ResNet-50 back- bone model is more accurate at identifying individuals in a variety of settings.

- Precision:92.6
- Recall: 89.3
- F1 Score:90.9

The model behaved stably under occlusions, different poses, and lighting changes, justifying its readiness to be deployed.

Violence Detection (MobileNetV2): A light-weight CNN optimised for mobile was used in frame-level violent vs. non- violent classification.

- Accuracy:87.4
- Precision: 85.9
- Recall: 88.1
- F1 Score: 87.0

The model's ability to perform high-speed inference makes it suitable for real-time video surveillance use cases executed on edge devices.

Weapon Detection (YOLOv5s): The *YOLOv5s* model, chosen due to its speed-accuracy compromise, was able to detect guns and knives.

- mAP@0.5:90.2
- Precision:89.8
- Recall:91.0

This model's ability to detect weapons in occluded or cluttered environments contributed significantly to the threat detection capability of the system.

CONCLUSION

This project demonstrates an end-to-end and scalable AI- powered public safety system combining:

- Faster R-CNN for precise person detection,
- MobileNetV2 for effective real-time violence classifica- tion,
- YOLOv5s for fast weapon detection performance, and

- Telegram Bot API to send real-time incident notifications.

A. Future Work

Future research will be focused on the following improve- ments to make the system more efficient, flexible, and effec- tive.

Facial Recognition and Behavioral Profiling: Facili- tating combination with facial recognition to recognize known offenders and behavioral profiling for early threat assessment through suspicious actions over time.

Multilingual Alerts and Interfaces: If working in mul- tilingual areas, the alerts should be shared in multiple languages.

Model Optimization for Edge Devices: The develop- ment of the optimized versions of the models (e.g., with quantization or pruning techniques) will enable more widespread use on power-constrained devices with little loss in detection accuracy.

REFERENCES

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ”You Only Look Once: Unified, Real-Time Object Detection,” *arXiv preprint arXiv:1506.02640*, 2015.
2. Z. Zhao, P. Zheng, S. Xu, and X. Wu, ”Object Detection With Deep Learning: A Review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
3. T.-Y. Lin et al., ”Microsoft COCO: Common Objects in Context,” in *Proc. ECCV*, 2014, pp. 740–755.
4. W. Liu et al., ”SSD: Single Shot MultiBox Detector,” in *Proc. ECCV*, The outcomes prove that the system is highly accurate 2016, pp. 21–37.*Proc.* and reliable in all its constituents, proving its use in smart surveillance of public areas.
5. subsectionLimitations While the proposed system per- formed promisingly, it has some limitations that need to be highlighted:
   - **Dataset Bias:** The training datasets might not encompass all the range of actual-world situations. For example, performance might deteriorate under conditions like low- light, fog, rain, crowd density, or non-standard camera angles that weren’t suitably represented in the dataset.
   - **False Positives:** The system might occasionally mislabel fast but harmless motion—like gestures in games, fun games, or unscripted crowd movement—as violent activ- ity and issue false alarms.
   - **Hardware Constraints:** Although the models run seam- lessly in real-time on GPU-based hardware or optimized edge hardware, the performance drops substantially when running on CPU-only hardware, so scalability in cost- constrained environments is restricted.
   - **Privacy and Ethical Issues:** Use of permanent video surveillance systems, particularly facial recognition or be- havior monitoring ones, poses ethical questions regarding the privacy, consent, and safeguard of individuals—above all, within public or semi-public spaces.
   - **Lack of Multimodal Sensing:** The current framework relies on visual data only from RGB video streams. The lack of supplementary modalities like audio (for verbal aggression detection) or thermal images deprives it of resistance in difficult settings.
6. K. He, G. Gkioxari, P. Dollar, and R. Girshick, ”Mask R-CNN,” in *ICCV*, 2017, pp. 2961–2969.
7. M. Everingham et al., ”The PASCAL Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

8. L. Liu et al., ”Deep Learning for Generic Object Detection: A Survey,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.
9. K. Simonyan and A. Zisserman, ”Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
10. C. Szegedy et al., ”Going Deeper with Convolutions,” in *Proc. CVPR*, 2015, pp. 1–9.
11. S. Ren, K. He, R. Girshick, and J. Sun, ”Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks,” in *Proc. NIPS*, 2015, pp. 91–99.
12. P. Viola, M. Jones, and D. Snow, ”Detecting Pedestrians Using Patterns of Motion and Appearance,” in *Proc. ICCV*, 2003, pp. 734–741.
13. N. Dalal and B. Triggs, ”Histograms of Oriented Gradients for Human Detection,” in *Proc. CVPR*, 2005, pp. 886–893.
14. C. Papageorgiou and T. Poggio, ”A Trainable System for Object Detection,” *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
15. S. Munder and D. M. Gavrila, ”An Experimental Study on Pedestrian Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1863–1868, 2006.
16. B. Wu and R. Nevatia, ”Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors,” *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007.
17. P. Dollar, C. Wojek, B. Schiele, and P. Perona, ”Pedestrian Detection: An Evaluation of the State of the Art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.
18. X. Zhang, L. Lin, and X. Wu, ”Fast Human Detection Using a Cascade of Histograms of Oriented Gradients,” in *Proc. ICIP*, 2006, pp. 149–152.
19. S. Walk, N. Majer, K. Schindler, and B. Schiele, ”New Features and Insights for Pedestrian Detection,” in *Proc. CVPR*, 2010, pp. 1030–1037.
20. M. Enzweiler and D. M. Gavrila, ”Monocular Pedestrian Detection: Survey and Experiments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.
21. A. Ess, B. Leibe, and L. Van Gool, ”Depth and Appearance for Mobile Scene Analysis,” in *Proc. ICCV*, 2007, pp. 1–8.
22. S. Khalid, A. Waqar, H. U. A. Tahir, and O. C. Edo, ”Weapon Detection System for Surveillance and Security,” in *Proc. ITIKD*, 2023, doi:10.1109/ITIKD56332.2023.10099733.
23. 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.
24. S. Jahan, R. Roknuzzaman, and M. R. Islam, ”A Critical Analy- sis on Machine Learning Techniques for Video-Based Human Activ- ity Recognition of Surveillance Systems: A Review,” *arXiv preprint arXiv:2409.00731*, 2024.
25. R. Chatterjee et al., ”Deep Learning-Based Efficient Firearms Monitor- ing Technique for Building Secure Smart Cities,” *IEEE Access*, vol. 11, pp. 37515 ::contentReference[oaicite:0]index=0
26. Viso.ai, “YOLOv8 Architecture Structure,” Online image, 2023. Available: https://viso.ai/wp-content/smush-webp/2023/12/ YOLOv8-Architecture-Structure-1012x1060.jpg.webp