Intelligent Systems with Applications 21 (2024) 200311

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Design of a real-time crime monitoring system using deep learning techniques

Md. Muktadir Mukto, Mahamudul Hasan, Md. Maiyaz Al Mahmud, Ikramul Haque, Md. Ahsan Ahmed, Taskeed Jabid, Md. Sawkat Ali, Mohammad Rifat Ahmmad Rashid, Mohammad Manzurul Islam, Maheen Islam *

Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh

ARTICLE INFO

Keywords:
Crime monitoring system (CMS)
Surveillance
Weapon detection
Violence detection
Face recognition
Deep learning
Image processing

ABSTRACT

Criminal Activities and Crime Monitoring System has long been a research topic. In this paper, we propose an effective Crime Monitoring System (CMS) that can detect a crime in real-time using a camera surveillance system and notify the appropriate law enforcement officer. The CMS was proposed to counterbalance human weaknesses such as inattention, slow reaction, and slacking, for example, in detecting crimes. The proposed CMS detects crime scenes by combining the mechanisms and functionalities of closed-circuit television (CCTV) cameras with various deep-learning methods and image-processing techniques. The CMS operates in three stages: weapon detection, violence detection, and face recognition. We used transfer learning models to detect weapons, and violence and used a face recognition algorithm to recognize faces. More specifically, to detect weapons YOLOv5, and MobileNetv2 to detect violence and used face recognition algorithms to recognize faces. The image dataset is used in CMS, while the video dataset is used to train the MobileNet-based violence detection model. In this case, frame-by-frame images extracted from video files were used to train the model. All of the models performed admirably. The weapon detection model detected four different weapon classes with greater than 80% accuracy. The violence detection model is also 95% accurate. The face recognition model had a 97% accuracy rate in detecting faces. The CMS's combined model was tested in a variety of real-world scenarios, and its performance was found to be outstanding. It was able to detect crime incidents and generate timely alarms, demonstrating its effectiveness in providing security and safety.

1. Introduction

The early use of surveillance cameras was to monitor individual actions by capturing live footage from the coverage area. Moving forward, new technologies and features, e.g., Infrared cameras, automated sensors, body temperature detection, night vision, advanced control mechanism, and 360-degree rotation were added to empower the system which is used as an indispensable means for many security solutions systems (Rai et al., 2018). Moreover, the use of Artificial Intelligence (AI) and Deep Learning plays a big role in improving the efficiency and reliability of video surveillance solutions for preventing crime and disaster (Sung & Park, 2021). Therefore, recent studies are focusing more on intelligent-based surveillance systems for disaster and crime prevention as well as industrial and public security protection. There are several proposed models for intelligent traffic video surveillance and accident identification (Hajri & Fradi, 2022) and enhanced algorithms for the resolution of intelligent video system images. However, few studies have explored video surveillance systems with cutting-edge technologies such as deep learning (Rasheed et al., 2014, Wei et al., 2018, Zhou et al., 2019, Xu, 2021).

Crime cannot be foreseen before it happens. However, the probable occurrence of a crime can be determined through carefully observing the behavior of the suspect and the surroundings. Nowadays, surveillance cameras are commonly used within an area to monitor individual activities for any abnormal or criminal incident. These surveillance cameras mainly transfer the captured videos over the IP networks that

* Corresponding author.
E-mail addresses: mdmuktadirmukto@gmail.com (M.M. Mukto), munna09bd@gmail.com (M. Hasan), mazmamud41@gmail.com (M.M. Al Mahmud), ikramulsohel101@gmail.com (I. Haque), tonmoy.ahsan000@gmail.com (M.A. Ahmed), taskeed@ewubd.edu (T. Jabid), alim@ewubd.edu (M.S. Ali), rifat.rashid@ewubd.edu (M.R. Ahmmad Rashid), mohammad.islam@ewubd.edu (M. Manzurul Islam), maheen@ewubd.edu (M. Islam).

https://doi.org/10.1016/j.iswa.2023.200311
Received 20 May 2023; Received in revised form 9 November 2023; Accepted 22 November 2023
Available online 7 December 2023

are viewed by the security personnel. Continuous and cautious monitoring is required by the security personnel to identify any unusual incidents timely and correct. This can be laborious and has deteriorating mental health effects. Thus, a crime incident can go unnoticed if the observer becomes careless or inattentive. To overcome these human weaknesses, an automated system is required to monitor human actions and predict the occurrence of a crime incident beforehand by utilizing ML and AI (Motiian et al., 2017).

To meet the expected result and efficiency in crime monitoring the system must go through a step-by-step process in a brief amount of time. An individual must meet some criteria to be determined as a suspect. The performance of the system depends on how correctly it determines if an individual is committing a crime or not based on these criteria. When two or more individuals interact with each other they sometimes show some behavioral patterns like handshaking, nodding, hugging, etc. While a person is aggressive the behavioral pattern extracted from the person can seem somewhat abnormal. Even when an individual is alone, his or her abnormality in behavior can be determined by how he or she interacts with the environment surrounding him. For example, a person who is about to steal something will not show the same behavioral pattern as a normal person. Identifying each of these behavioral patterns and matching them with the abnormal ones can be considered criteria. While committing a crime a person usually carries a lethal object, which in some cases can be considered a weapon. In a 3-dimensional space, every object has a structural pattern. Combining and comparing these patterns an object can be identified as lethal or non-lethal. In video files 3D monitoring is possible but in the case of 2D images, it is also possible to identify an object as lethal or non-lethal by combining and comparing the pixel frames. In a crime detection system, weapon detection can be another criterion. Another issue that can occur in crime detection is the ability to determine if the observed person is authorized to carry a weapon. While it is true that weapons can be used for committing crimes and breaking the law, law enforcers must carry weapons to keep the law and order. So, the system must be designed so that it can detect the authorized person and not interfere with their work. A face detection system can help to resolve this issue while also helping recognize convicts. These criteria can deliver enough information to the system for crime detection and a set of these criteria can refer to a problem the system must solve in its detection process

In this work, we have proposed a system that can detect crime beforehand through camera surveillance and notify the nearest law enforcers. The system is able to detect behavioral patterns and combine them with other gained information to determine the possibility of crime occurrence. The law enforcers notified by the system will be able to take action before the incident takes place. This will improve their performance and capability to deal with crime incidents efficiently. The contributions of this paper are summarized below:

- The utilization of Deep Learning to improve the efficiency and reliability of video surveillance solutions stands as a significant innovation. These technologies enhance crime and disaster prevention capabilities by enabling predictive analysis and automated anomaly detection.
- The innovative aspect of analyzing behavioral patterns to predict possible crimes is a departure from traditional monitoring. This AI-driven approach enables the identification of abnormal actions and behaviors that may indicate criminal intent.
- The development of an automated system that employs Deep Learning to predict crime incidents beforehand introduces a new level of efficiency and accuracy in crime prevention. This technology-driven approach has the potential to significantly enhance security measures.
- The use of a multi-criteria approach, including behavioral analysis, object detection, weapon identification, and face recognition, presents an innovative method for assessing and predicting criminal activities. This holistic approach offers a more comprehensive understanding of potential threats.
- The identification of suitable architectures, such as YOLOv5, and MobileNetV2, for the surveillance system demonstrates an innovative approach to selecting technologies that align with this paper's goals, leading to optimal performance and results.
- The innovation lies in the real-time analysis of video data from various sources. By processing video feeds in real-time and extracting valuable insights, the system contributes to rapid decision-making and effective response to potential threats.
- The inclusion of violence detection, weapon identification, and face recognition within the proposed CMS showcases an innovative approach to comprehensive crime detection. This holistic system ensures that multiple aspects of criminal activities are considered simultaneously.

In CMS the object detection task is carried out to detect weapons from image frames. As CMS is a real-time crime monitoring system the performance of the system heavily depends on efficiency and time complexity. By utilizing EffiecientNet architecture YOLOv5 can detect objects faster and by generating dynamic anchor boxes the model can shrink the point of interest by only covering the detected object. Another fast and efficient model that is used in this system is the MobileNetv2. MobileNetv2 utilizes the depthwise convolution layers in its architecture for faster classification tasks. It is a lightweight model that can be easily deployed. As such the model is used for abnormal behavior detection. Finally, the Local Binary Pattern Histogram (LBPH) is used for face recognition in CMS.

The remainder of the paper is organized as follows. A review of the literature is described in Section 2. The suggested methodology and the architectures used in the proposed CMS are described in Section 3 along with an analysis. Following that, in Section 4, the experiment findings are provided along with a discussion. Finally, Section 5 concludes the article.

2. Related works

Nowadays, many surveillance systems use deep learning techniques. In recent literature, studies have been conducted for weapon detection, face detection and recognition, abnormal behavior and anomaly detection, and human interaction recognition. This section summarizes some relevant research on machine learning and deep learning techniques connected to the work that went into this paper. In the case of weapon detection, the primary method is the concept of object detection and recognition using various CNN models. The reviewed papers suggested various approaches using several of these models such as VGG16, VGG19, ResNet, MobileNet, etc. The suitable object detection method is then selected for the weapon classification, recognition, and detection tasks. Face detection and recognition is a somewhat different and more complex task than weapon detection. Usual weapon objects have a defined pattern but every individual has a different facial structure. The papers reviewed for this purpose have given us a well-defined approach to handling this task efficiently. Abnormal behavior and human interaction detection tasks required extensive monitoring of deviation of actual data from sample data. The reviewed papers suggested methods to collect patterns and calculate deviations.

2.1. Weapon detection

In 2018, Navalgund and K. described a system that will help authorities to keep the incidents of crimes in the neighborhood in control through real-time videos and images by detecting crime incidents and informing them using VGGNet 19. In 2018, Wei et al. introduced a deep learning method for identifying and classifying people in video data that was collected using a high-power lens video camera from a distance of

several kilometers. In 2017, authors Verma and Dhillon describe a handheld gun detection system using faster R-CNN deep learning. In 2022, Mukto et al. describe a tool to classify different types of lethal and non-lethal weapons based on weapon characteristics such as weapon shape and size. Buckchash and Raman discuss their effort in 2017, which aims to develop object recognition algorithms for security cameras that can recognize particular items, such as sharp objects. In 2020, Alaqil et al. propose an Automatic Gun Detection system using the Faster R- CNN model. The algorithm in this study was developed in collaboration with the team that published (Grega et al., 2013) Automated Recognition of Firearms in Surveillance Video. This research presents the difficult issue of fully automated CCTV picture analysis and scenario detection.

2.2. Face detection and recognition

In 2019, Harikrishnan et al. discuss a visual attendance system in this research report. The system's four main components are face detection and data gathering, facial recognition training, facial recognition, and Excel-based attendance tracking. In this work, the vision was taught and evaluated in a classroom environment in a range of contexts, with a maximum recognition accuracy of 74% and above. In 2019, Ayed et al. suggest a new approach for detecting fear based on heart rate estimates. The suggested approach in this paper aims to enhance the performance of real-time heart rate estimates. The major goal is to extract the frightened expression from the face using a non-complex algorithm that combines the bandpass filter, a Lagrangian, and an Eulerian transformer. Here in 2021, the author Xu examines some of the most modern methods for object detection and face identification and explains why they could or might not be among the most effective for usage in video surveillance applications in terms of accuracy and speed.

2.3. Abnormal behavior and anomaly detection

In 2010, Takai discusses approaches to a detection method of suspicious activity from the observed person's behavior in a dynamic image in real-time and measures the degree of risk using the detecting branch position between suspicious and unsuspicious activity. The security camera system that analyzes behavior follows motion and focuses on the full or specified parts of the human body. In 2014, Rasheed et al. discuss tracking and detecting abnormal behavior in video surveillance using Optical flow and Neural networks. The proposed methodology mainly involves moving objects and tracking them afterward. As mentioned above the CMS must be able to identify every individual in a certain situation. Chumuang et al. (2018) describe an incident alarm tool based on CCTV cameras. The author's goal in this research is to create an automated alarm system based on a CCTV camera system while maximizing coverage efficiency. In 2019, the authors Wang and Xia propose a new approach for abnormal activity detection with DL features by combining (SDAE) with better dense trajectories. Identifying aberrant behavior through surveillance poses a significant challenge. Anomaly detection requires spare coding, and in this work (Zhou et al., 2019), a new neural network for anomaly detection is proposed by Zhou et al. by deeply achieving feature learning, sparse representation, and dictionary learning in three combined neural processing blocks. In Benito-Picazo et al. (2020), a Deep learning-based video surveillance system managed by low-cost hardware and panoramic cameras researchers Jesus Benito-Picazo et al. produced a video surveillance system for a panoramic 360-degree surveillance camera that detects moving objects with anomalous behavior. In order to build a video surveillance system with the least amount of hardware expenditure, the authors of the Benito-Picazo et al. (2020) study used a 360-degree camera and deep learning algorithms. The primary objective of Ramzan et al. (2019) is to give a thorough, comprehensive literature evaluation of the techniques for violence detection. SVM-based violence detection methods: The SVM Algorithm is reliable for binary classification problems since it takes into account numerical information.

2.4. Human interaction recognition

In 2019, Gong et al. emphasize how human Contact Recognition plays a crucial part in human-to-human interaction and the validation of interpersonal ties considering that it offers hard-to-extract information on a person's identity, personality, and psychological state. In 2018, Wei et al. introduced a deep learning method for identifying and classifying people in video data that was collected using a high-power lens video camera from a distance of several kilometers. The Adaboost person detector is used in this work to assess if there is a person in the moving regions after considering a rapid or computationally effective method for recognizing moving regions in an image.

From novel ideas to practical implementations, the papers contribute to the development of the CMS in meaningful ways. The papers collectively present fresh perspectives and novel techniques. Every paper has a well-defined problem and unique innovative approaches to solving those problems. The CMS comprises the problems that these papers address, which together form its fundamental challenges. For example, in a crime incident, CMS must detect a weapon and classify it while it also has to identify the person and observe his or her behavior. As these problems are merged into one it brings new challenges and issues that need to be addressed as well. Time complexity, interconnectivity, interdependency, and hierarchy of the problems are some of the challenges that must be faced while developing the system. For example, the system must be able to detect a person and identify him or her and afterward will have to detect if any weapon is wielded by him or her. Or, it can go the other way around. Or, it can be a simultaneous task. The hierarchy of the problems and their interdependencies must be addressed. Not to mention the system must execute these tasks quickly and efficiently to enable swift detection and monitoring of crimes. So, the papers provided individual solutions for individual and definite problems. As the problems are now merged the solutions must be modified or in some cases, completely new and unique solutions must be introduced so that the CMS performs as expected.

The Content Monitoring System (CMS) is composed of a collection of distinct modules, each tailored to address specific tasks. During the development of these modules, pertinent research and studies within the field were leveraged for comprehensive insights. These works encompassed proposals and discoveries that significantly enriched the overall comprehension of the tasks at hand and guided the formulation of fitting approaches. These related works collectively formed a knowledge foundation that propelled the CMS modules' development and led to more informed and effective solutions. Concise summaries of these pertinent related works are thoughtfully presented in the Table 1.

3. Proposed method

The CMS can be described as a complex classification system that contains multiple layers of binary and multi-classification tasks. The system continuously checks if a crime incident is occurring in real-time by processing image frames extracted from video feed through multiple classification tasks. The classification tasks include weapon detection, face detection, and abnormal behavior detection. While face detection and weapon detection can be considered multiclass classification, abnormal behavior detection is a binary classification problem as such different deep-learning NN models are used for various classification tasks.

In this work, we proposed an innovative approach to monitor and detect crime and alert the law enforcement authority. Our proposed system architecture has five layers as illustrated in Fig. 1. Section 3.1 to 3.5 describes the working principles of these layers in detail.

3.1. Sensor layer

This is the very first layer of our proposed system. Different visual sensors deployed in the environments capture real-time video of an incident in this layer and forward it to the video and image processing

Table 1
The summary of related works.

Citation	Proposed	Findings	Limitation
Navalgund and P. (2018)	Locate and identify crime incidents using VGGNet-19 and Faster RCNN. Sending SMS and restricting the SMS module.	The system acquired 100% accuracy of its objective. By utilizing the mentioned methods, crime incidents can be identified with efficiency.	The system has a tendency to flag inappropriate events as crime incidents. Cannot differentiate between a suspect and a law enforcer.
Gong et al. (2019)	Implementing a recognition system using methodologies - OpenCV, AlexNet network, LSTM (LONG SHORT-TERM MEMORY), Softmax, Deep network methods.	UT- interaction dataset results in 91.9% accuracy in the recognition. A combination of CNN+LSTM+HMM is more effective for the task	Time complexity. The proposed method uses many more resources sacrificing efficiency.
Harikrishnan et al. (2019)	Visual attendance system Using Haar cascade And LOCAL BINARY PATTERNS HISTOGRAMS (LBPH)	Maximum recognition accuracy of 74 percent	Recognition difficulties if image frames have more noise.
Takai (2010)	Observing a person's behavior and detecting suspicious activity using Motion recognition	Successfully identify suspicious behavior by comparing motion quantity and degree of risk	May identify an authorized person as a suspect
Rasheed et al. (2014)	Identifying normal or chaotic movement based on Gaussian mixture model, The optical flow and Feedforward neural network	The NN model has an accuracy of 97.5% and is most suitable for the task	The method may perform poorly in a crowded situation
Chumuang et al. (2018)	Developing an automated accident alerting system using OpenCV and Gaussian High-Pass Filter	Daytime accuracy of 80% and 50% at nighttime	Poor performance in less illumination
Wang and Xia (2019)	To efficiently detect and track abnormal behavior in complex and crowded situations using SDAE, SIFT flow, and Bandpass filter	The proposed method can detect abnormal behavior with 93% accuracy	Only detects human-to-human interaction. May fail to detect abnormal behavior in a situation where only one individual is present.
Ben Ayed et al. (2019)	To extract the fear feeling from the face using a non-complex algorithm such as Bandpass filter, Eulerian transformer, Lagrangian transformer, Haar Cascade and OpenCV	In heart rate estimation the system had an accuracy of 87%	
Wei et al. (2018)	Detect individuals in the video that is taken from a distance of several km using a highly capable lens camera. Classify the identified persons using Adaboost person detector, Self-defined CNN, AlexNet and GoogleNet.	CNN 83%, AlexNet 86% and GoogleNet 90%. Google Net outperforms other observed methods.	Could perform poorly in poor lighting
Zhou et al. (2019)	Developing a new neural network for anomaly detection and To minimize the challenge of sparse anomaly detection. Methods - Dictionary learning, RankSVM, CNN, LSTM	AUC curve 88.16%, Precision 92.26%, Recall 97.2%, True Positive 32, and False Alarm 8	
Verma and Dhillon (2017)	Described a detection system in which guns are handheld using VGG-16	Got the best result with 93.1% accuracy in detection.	May have difficulties identifying holstered gun
Buckchash and Raman (2017)	To create object identification algorithms in security cameras that can detect specific objects using FAST (Features from Accelerated Segment Test), MoG method for foreground detection and MRA (Multi-Resolution Analysis) method for classification	AST-FREAK combination yields a satisfying result of 96% accuracy	
Alaqil et al. (2020)	To propose an automatic Gun Detection system using CNN models such as ResNet50, Inception-ResNetV2, VGG16, MobileNetV2	Faster R-CNN with Inception-ResNet achieved an mAP of 81% with the lowest log-average miss rate	The training and test time is unsuitable for real-time applications
Benito-Picazo et al. (2020)	To detect moving objects with anomalous behavior Using CNN models.	Triangular uniform mixture model outperforms Gaussian-uniform distribution and Student-t distribution	Lags behind in system performance, while processing frame by frame
Xu (2021)	Developing an end-to-end video surveillance system that might be used as a starting point for more complex systems. Tools used -ResNet V1 on VGGFace2 with Multi-task Cascaded Convolutional Networks (MTCN)..	Best accuracy is achieved by training the model on a mixture of static images and image sequences.	Higher detection accuracies at real-time rates and may be more robust to difficulties such as occlusions.
Ramzan et al. (2019)	Presenting an exhaustive systematic literature review of the methods of violence detection.	Results reveal that the proposed method performs better than the different methods of handcrafted and DL	
Grega et al. (2013)	Creating a completely automated CCTV incident recognition using MPEG-7-based classifier.	Raise a flag when an uncovered firearm is carried.	There are still many wrong indications for a movie containing a lethal object.

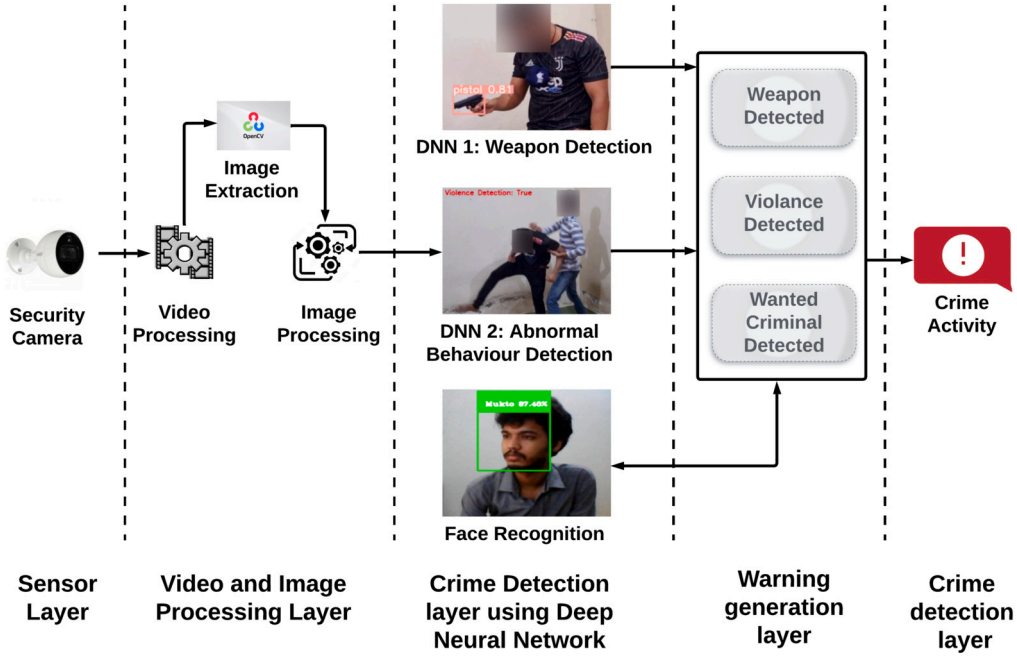


Fig. 1. System Architecture of Crime Monitoring System (CMS) using Deep Learning.

layer. High-definition close-circuit cameras are mostly used to capture video and images in real-time.

3.2. Video and image processing

The surveillance camera's video stream may include multiple image frames. While the surveillance and monitoring process is carried out in real time, each frame from the live camera stream is independently captured. Before moving on to the next layer, the detection layer, the image frames first undergo preprocessing, which includes scaling, resizing, brightness enhancement, and sharpening. To optimize the model's performance, addressing issues arising from the disparate sizes and proportions of input image frames is crucial. To ensure uniformity, these frames undergo resizing and rescaling during the preprocessing stage. Images captured in low-light conditions, a common challenge in dark environments, are subject to brightness enhancement techniques. This refinement serves to enhance visibility, facilitating the identification of areas of interest within the image frames. Furthermore, to enrich the diversity of the training dataset, a thoughtful approach to data augmentation is employed. This involves flipping and applying various transformations to the images, presenting the model with scenes captured from different angles. The goal is to foster the model's adaptability to a range of perspectives, ultimately enhancing its robustness. In certain instances, normalization techniques are judiciously applied to scale pixel values within a standardized range derived from a general minimum and maximum. This normalization ensures that the model is less sensitive to variations in pixel intensity, contributing to improved convergence during training. Through this comprehensive preprocessing pipeline, the model is better equipped to discern patterns and features within image frames, promoting more accurate and reliable performance across diverse scenarios. CMS is an ensemble model that utilizes three different methods to detect crime. One of them is the weapon detection model YOLOv5. The model is trained with a weapon dataset that contains weapon images of various sizes and shapes. Image frames are extracted from video streams. The trained models are applied to the extracted frames to detect crimes. If a weapon exists in that image frame, then the weapon is isolated by the model using a rectangle-bound box. Labeled data for all weapon types is stored as YAML files. The YOLOv5 model utilizes the labeled data to properly detect the weapon and isolate it from the rest of the image frame.

Algorithm 1 Pre-processing algorithm.

```
Require: Weapon image and violence video
  Begin
  Provide weapon image
  Provide violence video
  for each saven sample image of train and test do
    Rescale the image
    Rotate the image
    Do horizontal and vertical flip
  end for
  for each video file do
    Extract frames from the video
    for each saven sample image in the video do
      Rescale the frame image
      Random brightness of the frame image
      Rotate the frame image
      Do horizontal and vertical flip
    end for
  end for
  End
```

At the beginning of the Crime detection procedure lies the crucial step of image extraction and pre-processing, laying the groundwork for model training. To equip the model with a fitting dataset encompassing both images and videos, a methodical curation process is paramount. Specifically, every 7th image undergoes careful image processing applications, encompassing rescaling to enforce uniformity in size and form, as well as horizontal and vertical flips to augment dataset diversity. In the case of video data, frames are initially extracted from the sequences and subjected to analogous image processing procedures. This comprehensive pre-processing approach ensures a seamless treatment of both individual images and video frames, establishing a robust foundation for an adept crime detection model.

3.3. Crime detection layer using deep neural network

Three deep-learning models make up the criminal detection layer, as was previously described. Each model makes a contribution to the process of crime monitoring and detection. The YOLOv5 object detection model will be utilized as the initial model for weapon detection. Convolution layers and other features make this model a good choice for object detection in image frames, especially when they are familiar objects like guns. Another CNN model utilized in this system is Mobilenet

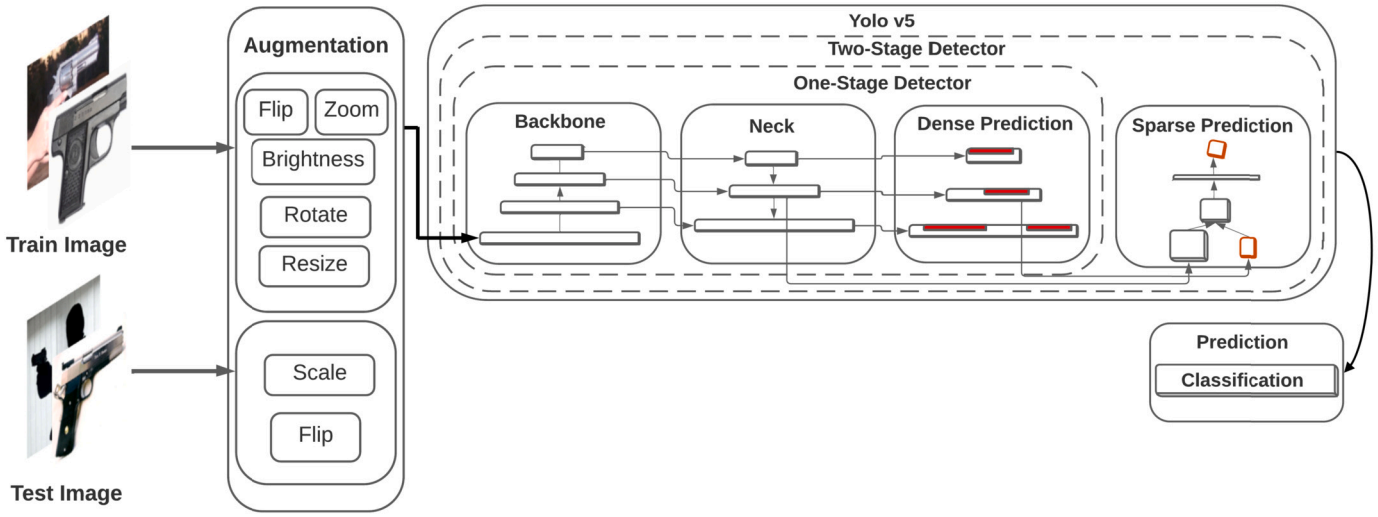


Fig. 2. Weapon detection process using the YOLOv5 object detection.

V2. The suspect and his surroundings must be scrutinized in order to spot aberrant conduct. As a result, in order to draw a conclusion, every facet of the suspect's behavior must be carefully examined. Mobilenet V2 is a quick and effective solution for this. The final model is combined with the Local Binary Pattern Histogram (LBPH) to extract facial traits. Afterward, prospective suspects or law enforcement are found and identified using the extracted features. All the building blocks of the system architecture are discussed in the following sections 3.2.1, 3.2.2, and 3.2.3.

3.3.1. Weapon detection

The task of weapon detection involves multiple classes. The model must reliably identify the type of weapon in addition to detecting its existence. Many photos of various weapons, including pistols, knives, rifles, and more, can be found in the weapon database and used to train the model. This makes it easier to determine if the detected weapon is lethal or not. With CMS, finding weapons comes first because it can ultimately establish whether or not a crime incident is happening.

We used the object identification algorithm YOLOv5 to find weapons. YOLOv5 enables users to quickly move from model training to application development. Model configurations and class values are stored in YAML file formats, which are used by the YOLOv5 model. YOLOv5 quickly and effectively predicts and recognizes things of interest in an image. The model augments itself throughout training and doesn't need the user's input anymore. The backbone, neck, and head are the three components that make up the model architecture.

YOLOv5 object detection architecture is displayed in Fig. 2. Features from an image are retrieved here using the backbone. Pyramids of feature data are produced through the neck. Using feature pyramids, the same visual objects with various attributes can be recognized. With the help of the data in the YAML file, the head can identify items in an image. ReLU and Sigmoid functions make up the activation function. ReLU is utilized in the hidden layers, and Sigmoid is used in the output-producing final prediction layers. The train and test data were divided in a ratio of 0.8 to 0.2 in order to train the model. The basic image file (PNG, JPG, etc.) and the XML file, which contains specific picture data such as image size, scale, label, etc., are both present in the dataset directory.

There are 242 photos in the validation set and 1072 in the training set. The YAML files are generated after the XML file and image files are obtained. The classes or types of objects are contained in the YAML file. It primarily functions as a prediction tool. Knife, grenade, handgun, and assault rifle are the distinguishable weapon classes. A batch of 16 and 30 epochs is used to train the model. In YOLOv5, the weights of the photos are decided. The model is then used for object detection after the training procedure is finished. On a test dataset of 0.2 with a confidence value of 0.3 and previously established picture weights, the detection technique is used. The results of the detection process are to be discussed in the experiment and result section given below.

Algorithm 2 Weapon Detection Algorithm.

Require: Extracted and Pre-processed image
Begin
Set image size
Set batch size
Rescale the image
Do augmentation
Label the images
Make yaml file for the label images
Initializing the pre-trained YOLOv5 model with weights and input
Set hyperparameters
Set optimizer
Train the model with the required image
fit the model with epoch number
Save the model
End

In the realm of crime detection, an indispensable module revolves around weapon detection and classification, with the robust YOLOv5 model at its core. Commencing with a methodology akin to abnormal behavior detection modules, the process involves image processing and augmentation. Key to YOLOv5's efficacy is the utilization of labeled images for data. YAML files, serving as meticulous configuration files for YOLOv5, dictate the nuances of its weapon detection procedures. These files intricately outline how YOLOv5 should execute its tasks. The YAML data file, a cornerstone in this endeavor, encapsulates vital information regarding the labeled image data that will be employed to train the model. Following the establishment of initial inputs and weights, hyperparameters, and optimizer settings to govern the model's performance, the YOLOv5 model undergoes training with a specified number of images, ensuring its prowess in weapon detection is honed to the task at hand.

3.3.2. Abnormal behavior detection

Even without firearms, criminals can still commit crimes. Since it can occur anywhere, at any time, and for any reason, unarmed violence can be regarded as the most prevalent type of crime. Typically, violence can be defined as an aggressive interaction between two or more people. Therefore, it is necessary to observe the interactions between two or more people in order to detect violence.

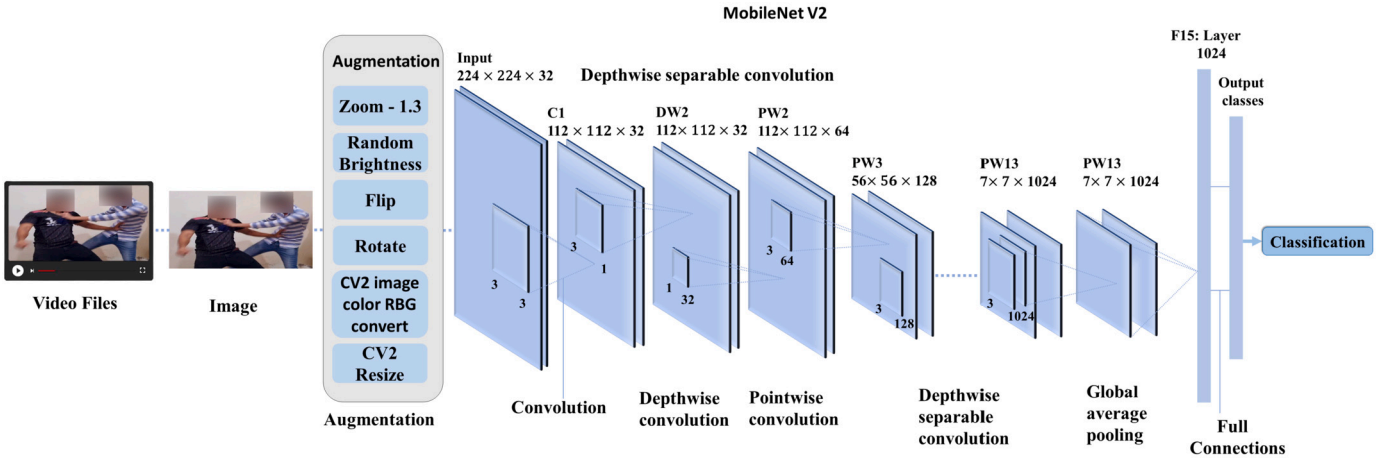


Fig. 3. Abnormal Behavior Detection process using the MobileNetv2.

MobileNetV2 was employed in this particular scenario. A convolution neural network with 53 layers is called MobileNetV2. A dataset including hundreds of image data can be quickly and effectively classified using MobileNetV2. The network includes photos with 224x224 image sizes. This can be applied to simple image categorization tasks.

The architecture of MobileNetv2 which is shown in Fig. 3, begins with an initial fully convolutional layer featuring 32 filters. This is succeeded by 19 residual bottleneck layers. To introduce non-linearity in a low-precision computation setting, ReLU6 is employed due to its resilience. The conventional kernel size of 3 × 3 is consistently adopted, aligning with contemporary network norms. Additionally, during the training process, dropout and batch normalization techniques are incorporated.

Prior to sending the final data set to the output, the depthwise separable convolution is employed to reduce the number of parameters. It is made up of several depths and pointwise convolutional layers. Each input channel is simultaneously convolved with the filter's depthwise convolution to create the output channels. The final output map is created by passing the output of a depthwise convolution through a pointwise convolution and applying a 1×1 filter. Depthwise separable convolution is used to maintain a small number of parameters in order to reduce time complexity and increase the efficiency of the model.

Depthwise Separable Convolutions are fundamental components in various efficient neural network designs, including our current study. This technique involves substituting a standard convolutional operation with a two-step approach. The first step entails a depthwise convolution, which employs a solitary convolutional filter for each input channel, thereby conducting lightweight filtering. Following this, a 1 × 1 pointwise convolution serves as the second step, generating fresh features by calculating linear combinations of input channels. This strategy optimizes computation and enhances the efficiency of neural networks while preserving essential information. Depthwise separable convolutions significantly decrease computational requirements in contrast to conventional layers, by a factor roughly approximated as k times 21. In the context of MobileNetV2, where k is set to 3, signifying the utilization of 3 × 3 depthwise separable convolutions, the computational overhead is reduced by a factor of approximately 8 to 9 when compared to standard convolutions. This substantial reduction in computation comes with only a minor sacrifice in accuracy.

Computational Cost of Depthwise Separable Convolutions:

$$hi \times wi \times di(k^2 + dj) \tag{1}$$

Here, hi and wi represent the pixel height and width while di is the dimension. The total count of multiply-add operations necessary for a block with dimensions h × w, an expansion factor of t, a kernel size of k, d′ input channels, and d″ output channels can be computed as:

Table 2
Bottleneck residual block transformation.

Input	Operator	Output
h × w × k	1x1 conv2d , ReLU6	h × w × (tk)
h × w × tk	3x3 dwise s=s, ReLU6	h/s × w/s × (tk)
h/s × w/s × (tk)	linear 1x1 conv2d	h/s × w/s × k

$$h \times w \times d' \times t(d' + k^2 + d'') \tag{2}$$

The transformation of the k channel to k′ channel with stride s, and expansion factor t through residual blocks can be described in Table 2

In Table 2 the first 1*1 convolution layer pointwise kernel size of 1 is applied. Here the activation function is ReLU6. The terms h and w represent the dimensions of the image which are height and weight. The term k represents the number of output channels. Expansion factor t is added to the output channels. If the input image has 64 channels and the expansion factor is 6, then the output channel would get 64*6 = 384 channels. The stride is a hyperparameter that determines the step size at which the convolutional filter moves when scanning the input data. It is denoted by 's' in a depthwise convolution layer. If we have a depthwise convolution with "w/s * h/s * tk", it means that the depthwise convolution filter is a w*h filter, and it moves with a stride of 1 pixel in both the horizontal and vertical directions. Hence for stride 1, the filter will not skip pixels.

The dataset contains videos labeled "Violence" and "Non-violence". The video files are put through MobileNetV2 layers to train the model. To obtain the output a dense layer has been used with the sigmoid activation. This serves as the input for the next layer. The output value always ranges between 0 and 1. The callback is used to end the training when the required values of validation are reached. The model is trained through epoch 100. The learning rate(LR) parameter is set between 0.00001 and 0.00005. The size of the batch is 4. The early stop of the model training will activate when 0.999 accuracies are gained with verbose 1 and min delta 0.00075. The min delta parameter defines the change in accuracy while training the model. To initiate the prediction process the training and testing data are split into 0.7-0.3. When a video is given as input the system tries to predict violence status by matching input data with labeled frames. As every frame in a video is checked thoroughly, the system will identify violence in the video when only those frames are on screen. While training the best epoch can be determined by the validation loss value. The test results are discussed in the result section.

The Violence Detection model stands as a pivotal component within the CMS modules of the crime detection layer, employing MobileNetV2 for its specialized capabilities. At the onset, crucial parameters such as image size and batch size are defined to govern the input frames fed

Algorithm 3 Violence Detection Algorithm.

Require: Extracted and Pre-processed image
Begin
Set image size (128, 128)
Set batch size
Initializing the pre-trained MobileNetV2 model with weights and input
Set activation as "sigmoid"
Set layers trainable false
Compile the model use [loss = 'categorical crossentropy', optimizer = 'adam', metrics = ['accuracy']]
fit the model with epoch number
use EarlyStopping to stop the training process as soon as the monitored metric stops improving
use ModelCheckpoint to save the model
End

into the model. The activation function is strategically set to 'Sigmoid,' a well-suited choice for binary classification tasks in convolutional neural networks (CNN). In the model compilation phase, key configurations are specified, including cross-entropy loss for effective classification, the adaptive learning prowess of the Adam optimizer, and the performance metric set to accuracy. Moreover, a prudent approach is taken in model preservation, with intervals designated for periodic model saving. To optimize training efficiency, an early stop mechanism is incorporated, poised to halt the training process promptly when the model exhibits no discernible improvement in performance metrics. With all these parameters and strategies in place, the model undergoes fitting with a defined number of epochs, fine-tuning its capabilities for robust violence detection within the crime detection framework.

3.3.3. Face recognition

Once more, it is true that law enforcement personnel can possess guns. In order to identify whether or not the individual carrying a firearm is a law enforcement official, the system will use face recognition. The system will disregard the occurrence if the person is a law enforcement official by using a database of the law enforcement official. Further inspection will be done if the person is not a law enforcement officer.

In this paper, the Local Binary Pattern Histogram was used for face recognition and face recognition architecture is shown in Fig. 4. First face detection is done using deep learning. An image is provided which is used to detect the exact point where the face is located in the image, facial landmark and the facial pattern are used for the procedure. Using face alignment, the geometric structure of the face is extracted. After pointing out the face in the image the image is cropped to the content. The cropped image is passed to the recognition block. FaceNet DL is used to calculate a 128-D embedding. The 128 embeddings are utilized in triplet loss to train the model. Each input batch of data includes three images. Anchor, Positive, and Negative. The main 128-D embeddings are the anchors. The embeddings of anchor and positives stay close while negative embeddings are pushed further away. Thus, embedding helps to train the neural network.

To obtain face data, the model takes multiple video feeds as input. The image frames are taken from the video feed and converted to grayscale. For every input from 1 to n, the Haar Cascade classification is used to detect a face in a frame image and crop it to that region. Haar Cascade classification is a machine-learning object detection method used to identify objects in images or videos. Haar-like features are simple rectangular filters that are applied to an image to capture various patterns of light and dark regions. These features can represent edges, corners, or other textures. The Haar Cascade classifier is trained using positive and negative images. Positive images contain the object of interest, and negative images do not. Adaptive thresholding is used to ignore non-object regions and crop the image frame to an object region. The cropped image is saved for further processing. A unique ID will be assigned for every individual from 1 to n detected from the video inputs. For training purposes, individual images must be provided by the operator. In this case, through preprocessing, the image is scaled and resized appropriately. Then the image is changed to grayscale pixels, cropped to content, and saved for training and testing.

Now to train the model, a Local Binary Pattern Histogram is used. It is a computer vision method that divides an image into multiple smaller regions. It then extracts texture descriptors from these smaller regions based on local binary patterns. LBP is calculated by comparing the intensity of a central pixel with its neighboring pixels. For each pixel in an image, a binary code is generated based on whether the neighboring pixels are greater than or equal to the central pixel's intensity. From the training dataset, the images are obtained, and for each image patch, an LPB histogram is constructed. The histogram bins correspond to different LBP patterns.

Finally, for prediction image frames are taken from the video stream, converted to grayscale, and then the system will recognize the face from the frames. During the face recognition process after the given frames are taken through the process, the system will deliver a predicted image ID with a confidence value. If the confidence value exceeds a threshold value the face will be recognized as that ID holder. For instance, a person in the dataset containing id 1303 and the threshold value is 50. After providing an image frame and running the prediction process if the confidence of id 1303 is greater than 50 then the person in the image will be recognized as id 1303. Otherwise, the person will not be recognized.

Algorithm 4 Facial Recognition Algorithm.

Require: Video File
Begin
Extract frames from the video
Detect face from the frame using the Haar Cascade Classifier
for Do the process until the video ends **do**
  if Face is found **then**
    Level the image with a unique ID
  end if
end for
Convert the frames RGB to grayscale image
Save the images with their label
Train the extracted images using LBPH face-recognize
Save the model
End

Haar Cascade classifier is utilized to detect faces in these image frames, applying a rectangular filter to focus on the areas of interest while excluding non-object regions. Once a face is identified, a distinct ID is assigned to the individual. To train the model, the grayscale version of the image is employed using the LBPH method, enhancing the model's ability to recognize facial patterns effectively. This methodology ensures a meticulous and accurate facial recognition process within the system.

3.4. Warning generation layer

When an incident occurs in the coverage area of a CCTV camera the incident is recorded. It is important to take important information from the recorded incident such as facial features, behavioral patterns, and the presence of a weapon, and store it in a database for further analysis. The facial features are used to identify, and track suspects. The time and form of the incident would also be recorded. Information extracted from the incident can help to analyze the frequency of crimes taking place, the effectiveness of any measures taken to reduce the crime rate, and the general behavior of people in a certain environment.

The messages that the CMS generates based on observed incidents can be categorized as follows: (i) The Weapon detected is generated by the system when any individual is detected carrying a lethal weapon and this detection can be defined as Dw. (ii) Violence detected is generated by the system when any form of violence is taking place such as fist fighting, riots, etc. and this detection can be defined as Dv. (iii) A wanted criminal detected message is generated by the system when the system is able to identify a wanted criminal based on the data provided

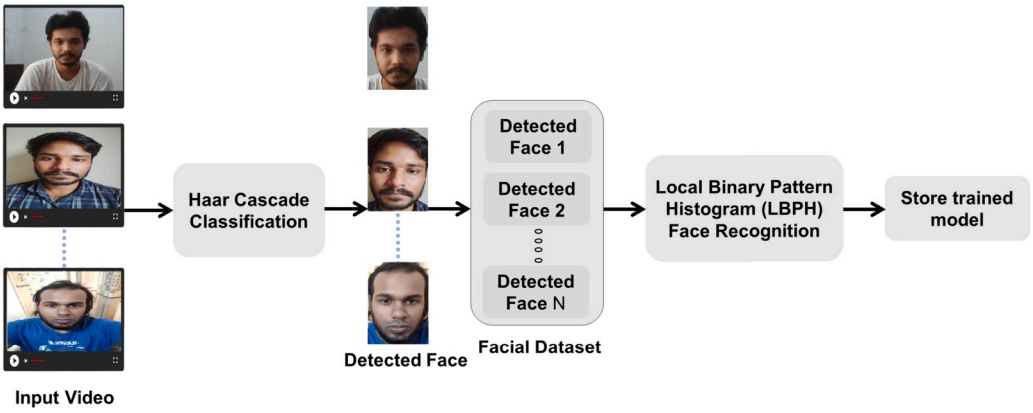


Fig. 4. Face recognition architecture.

via the suspect list. As depicted in the figure, a bi-directional interaction between the message layer and crime detection layer occurs during the face recognition and identification process. For example, the system may try to detect a convict before or after labeling an incident as a crime incident and this detection can be defined as Dwc. (iv) Criminal activity is generated by the system when the system is able to identify a crime incident by evaluating all the criteria mentioned before. Now, we can determine the criminal activity from the following equation,

$$C_a = D_w \cap D_v \cap D_{wc} \tag{3}$$

where Ca is the criminal activity, Dw is weapon detected, Dv is violence detected and Dwc is wanted criminal detected.

Finally, (v) Observation required is generated by the system when an incident is not considered a crime incident but has all the elements necessary to cumulate into one. Now, we can determine the observation required by using the following equations,

$$O_r = D_{wc} \cap (D_v \cup D_w) \tag{4}$$

or,

$$O_r = D_v \cap (D_{wc} \cup D_w) \tag{5}$$

3.5. Crime detection layer

A typical alert system comprises many levels of alerts, each of which corresponds to a distinct level of severity and urgency. For instance, the lowest alert level in a security system can denote a minor problem, whereas the highest alert level might denote a serious security violation. Similarly, the CMS's alerts based on observed occurrences may be divided into the following categories: (i) Danger - Extreme, (ii) Warning- High, (iii) Caution- mid, and (iv) Observe - low. The purpose of the CMS alert levels is to give law enforcement officials clear, succinct information so they can determine the urgency and gravity of a situation and take the necessary action. The 'Danger' alert is generated when criminal activity is imminent. If quick action is not taken, the operators and authorities can cause casualties. A 'Warning' alert is generated when criminal activity is not imminent but may occur if action is not taken. When this alert is given law enforcer must make a decision fast. The 'Caution' alert is generated when there is a low-level risk of a crime incident. This incident may require full attention from the operators. The 'Observe' is the lowest level alert that is generated when an incident can not be identified as a criminal incident but observation is required from the operators. In this case, operators may take their time to observe and accept their decision.

Algorithm: Alert generation Implementation. Input: (i) WD - Weapon detected, (ii) FD - Face detected, (iii) FI - Face Identification, (iv) ABD - Abnormal behavior detected, (v) IPETO - Individuals present except the observed. Require: Alert.

Algorithm 5 Alert generation Implementation.

```
Require: (i) WD - Weapon detected, (ii) FD - Face detected, (iii) FI - Face Identification,
  (iv) ABD - Abnormal behavior detected, (v) IPETO - Individuals present except the
  observed.
  Begin
  for each input WD, FD, FI, ABD, and IPETO do
    if (WD and FD and ABD and IPETO is True) and (FI is Suspect or Law Enforcer) then
      set alert = = Danger
    else if (WD and FD and ABD is True) and (IPETO is False) then
      if FI is Suspect then
        set alert = = Warning
      else
        set alert = = Caution
      end if
    else if (FD and ABD is True) and (WD is False) and (FI is Suspect) then
      if IPETO is True then
        set alert = = Warning
      else
        set alert = = Caution
      end if
    else if (WD and ABD is True) and (FD is False) then
      set alert = = Danger
    else if (WD and FD is False) and (ABD is True) then
      if IPETO is True then
        set alert = = Caution
      else
        set alert = = Observe
      end if
    else if (WD is False) and (FD is True) then
      if (ABD is False) and (FI is Suspect) then
        set alert = = Observe
      else if (ABD is True) and (FI is Law Enforcer) then
        set alert = = Observe
      end if
    end if
  end for
  End
```

Table 3 shows the alert levels of CMS. The CMS (Crime Monitoring System) is equipped to detect crime incidents and generate alerts based on certain criteria that indicate the severity of the situation. These criteria include (i) WD - Weapon detection, (ii) FD - Face detection, (iii) FI - Face Identification, (iv) ABD - Abnormal behavior detection, and (v) IPETO - Individuals present except the observed. The severity of the situation is determined based on these criteria. For instance, if an identified suspect is detected with a weapon and exhibiting suspicious behavior while other people are present, the situation would be considered highly severe. However, if the suspect is alone and unarmed, the situation would be less severe. The criminal detection system is improved in accuracy and efficacy by using these criteria.

Table 3
CMS alert levels.

WD	FD	FI	ABD	IPETO	Alert
True	True	Suspect	True	True	Danger
True	True	Suspect	True	False	Warning
False	True	Suspect	True	True	Warning
False	True	Suspect	True	False	Caution
False	True	Suspect	False	False/True	Observe
True	False	-	True	True	Danger
True	False	-	True	False	Danger
False	False	-	True	True	Caution
False	False	-	True	False	Observe
True	True	Law Enforcer	True	True	Danger
True	True	Law Enforcer	True	False	Caution
False	True	Law Enforcer	True	True/False	Observe

4. Experiments and results

As the system consists of multiple models, the results and performance of all the models have been highlighted along with the performance of the application as a whole.

4.1. Description of datasets, experimental setup, and data preprocessing

The datasets, experimental setup, and data preparation covered in the next subsections are used to evaluate our proposed system.

4.1.1. Description of datasets

This system requires photographs and videos of a variety of weapons, crime scenes such as pointing weapons, fighting, and so on, as well as photos of people's faces to recognize them. Based on this, a well-prepared dataset has been sought on the internet for this system to operate successfully, but no such image or video dataset has been obtained. As a result, a bespoke dataset was developed for this CMS to properly detect crime. There are 3114 images and videos in the dataset. The dataset can be divided into several categories such as Weapon Detection: 1314, Violence Detection: 800, and Face Recognition: 1000.

Since a quality system requires a decent dataset to function effectively, a custom dataset based on photographs and frames from the video has been constructed for the CMS. For the purpose of performing the weapon detection function, first, images or video frames of lethal weapons, such as firearms, knives, grenades, and assault rifles, were taken from the internet. For the purpose of detecting crimes, photographs and video frames from websites like Kaggal, Google Images, and a few other websites were used. These images showed people pointing weapons at people or attacking them, holding weapons in their hands, or directing them in the direction of the victim. As part of the algorithm training for the face recognition function, the authors provided their face images. This is the process used to build the dataset for this system.

4.1.2. Experimental setup

Since this investigation focuses on video observation, video records will contribute to the framework. As a result, some in-depth learning models using images should be constructed as the video will eventually be replaced entirely by picture outlines for differentiation. A picture dataset was created for profound learning models to use, and the analysts have provided that picture dataset. The machine's hardware specifications for the test were as follows: a 128 GB SSD, an Intel Core i7 CPU with two speeds of 1.8 GHz and 1.99 GHz, and 8 GB of RAM.

4.1.3. Data preprocessing

As three models were developed, the datasets required for the models were also different. As such the dataset preprocessing for different models is discussed separately.

The images were collected from the internet and Kaggle to prepare the dataset. This dataset contained 5000 images. Many faulty images were discarded and the final dataset contained 1314 images.

The violence detection dataset obtained from Kaggle initially contained 2000 videos. After random selection, 800 videos were used for the model. The videos are divided into two classes, Violence, and Non-Violence class with each class containing 400 videos. Then during the training phase, each video has been converted to an image frame by frame, which provides a large collection of images to train the model.

The face recognition dataset was created from the images provided by the authors. Some images where the facial features were unidentifiable due to blurriness, less lighting, and different angles were discarded.

4.2. Model validation

In Fig. 5 the training loss and validation, loss have been visualized through some graphs. The performance of a model can be determined by comparing two graphs. The model overfits if validation loss decreases and increases again. It is underfitting if the validation loss is too high. A perfectly aligned figure indicates the performance of the model is perfect.

In this model, there are 100 epochs which are the object loss, box loss, and class loss of training and validation datasets that align very well with some differences. The deviation is not very high in all the cases. In this case, the model will perform very well with both training data and new data.

4.3. Performance metrics

We evaluate our proposed method using the following performance measures.

4.3.1. Accuracy

Accuracy is defined as,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{6}$$

where FP (False Positive) is the number of positive images that are incorrectly classified as negative, FN (False Negative) is the number of negative images that are incorrectly identified as positive images, TP (True Positive) is the number of negative images that are correctly detected as negative images and TN (True Negative) is the number of positive images that are correctly detected as positive images.

4.3.2. False negative rate (FNR)

A low False Negative Rate (FNR) is a need for any CMS in order to ensure system security. FNR is provided by,

$$FNR = \frac{FN}{TP + FN} \times 100\% \tag{7}$$

4.3.3. Sensitivity

Sensitivity is a measurement of how many violent images are really appropriately classified as such. It comes from,

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

4.3.4. Specificity

The percentage of real images that are accurately identified as such is known as specificity. It comes from,

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{9}$$

4.4. Results and discussion

The following subsections discuss the results of our proposed system.

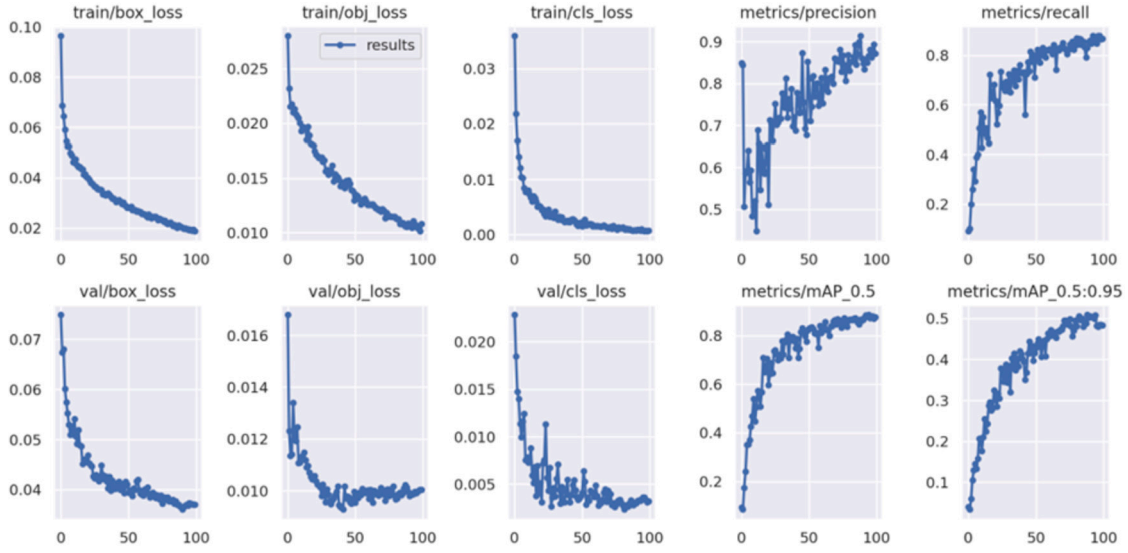


Fig. 5. Training loss and Validation loss.

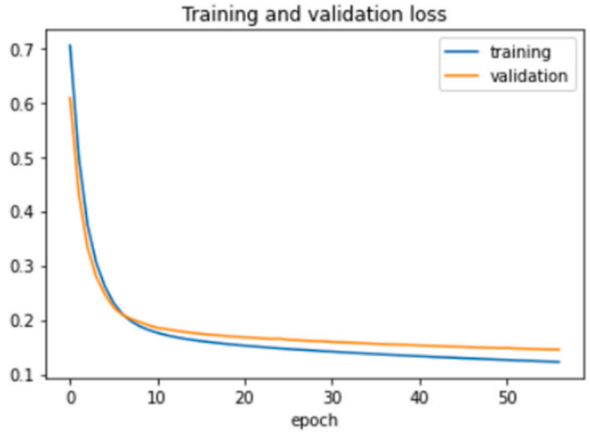


Fig. 6. Training and Validation loss plot for violence detection model.

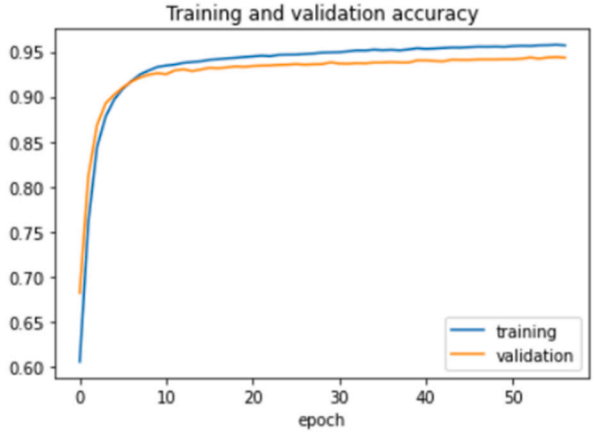


Fig. 7. Training and accuracy plot for violence detection model.

4.4.1. Violence detection

The first result and performance evaluated here are the model for violence detection that uses MobileNetV2. The model runs within an epoch range of 100 and the performance of the model can be determined by the number of epochs it is required to get the best accuracy during prediction along with the best accuracy achieved. The last validation loss change was observed in epoch number 65. The best epoch is 57. The early stop was triggered in epoch 67. The best epoch can be calculated from the changes in accuracy and validation loss.

Figs. 6 and 7 show the change in accuracy and validation loss in each epoch until the best epoch is reached.

The model's performance improves throughout training and validation as accuracy increases and validation loss decreases as it refers to how much the actual result is diverted from the predicted result. It can be said that the model performed well as it required nearly half the number of epochs to reach the best prediction result. After epoch 57 the change in validation loss and accuracy are bare minima and negligible. It can be observed that in both the loss and accuracy graph, the training model somewhat performs better than the validation model. This means in some cases the model may show some unexpected results. As this difference is not that significant it can be ignored.

Now we evaluate the model based on how many correct predictions are achieved. The prediction parameter is set to 0.5. The parameter can be changed but an increase or decrease in the parameter may have resulted in a false-positive and false-negative respectively. Out of all the data: correct predictions: 4375 and wrong predictions: 254.

Table 4
Violence detection model performance summary.

	Precision	Recall	F1-Score
Nonviolence	0.94	0.93	0.93
Violence	0.95	0.96	0.95
Weighted avg	0.95	0.95	0.95
Macro avg	0.94	0.94	0.94

The precision parameter in Table 4 describes the correct predictions of positive observations, while recall defines the correctly predicted positive observations in that class. The f-1 score provides an overall observation that takes precision and recall into account. As we can see, all of the parameters produce a high value that is always greater than 0.9. This indicates that the model performed as expected, with very low false positive and false negative rates.

In the context of Sumon et al. (2020), the authors introduce various deep-learning methodologies with diverse parameters aimed at detecting violence in video streams. The evaluation of these models involves a comparative analysis with the MobileNet V2 model, utilized in the violence detection phase of the CMS, to assess performance. Notably, the dataset employed by the authors comprises labeled video files categorized as violent and non-violent based on content, aligning closely with

Table 5
Violence detection performance comparison.

Models	Precision	Recall	F1-Score
Transfer with freezing	0.97	0.86	0.91
Transfer without freezing	0.97	0.92	0.94
VGG16 + FCN	0.95	0.87	0.90
VGG19 + FCN	0.94	0.89	0.91
ResNet50 + FCN	0.94	0.98	0.98
VGG16 + LSTM	1.00	0.91	0.93
VGG19 + LSTM	1.00	0.95	0.98
ResNet50 + LSTM	0.95	1.00	0.98
VGG16 + attention	0.96	0.87	0.90
VGG19 + attention	0.93	0.87	0.89
ResNet50 + attention	1.00	0.92	0.96
MobileNetV2 (CMS)	0.95	0.96	0.95

Table 6
Non-Violence detection performance comparison.

Models	Precision	Recall	F1-Score
Transfer with freezing	0.89	0.98	0.93
Transfer without freezing	0.95	0.99	0.97
VGG16 + FCN	0.91	0.96	0.93
VGG19 + FCN	0.90	0.97	0.93
ResNet50 + FCN	0.98	0.94	0.97
VGG16 + LSTM	0.87	1.00	0.93
VGG19 + LSTM	0.93	1.00	0.97
ResNet50 + LSTM	1.00	0.94	0.97
VGG16 + attention	0.85	0.96	0.90
VGG19 + attention	0.86	0.93	0.89
ResNet50 + attention	0.91	1.00	0.95
MobileNetV2 (CMS)	0.94	0.93	0.93

the dataset used to train the CMS MobileNet V2 model. To evaluate the efficacy of the CMS violence detection model in classifying positive and negative data, its evaluation metrics are compared with those of the models proposed by the authors. Performance metrics for the detection of positive outcomes are detailed in Table 5, while Table 6 provides insights into the models' performance in detecting negative outcomes. This comparative framework offers a comprehensive understanding of how the CMS MobileNet V2 model aligns with alternative deep-learning approaches in violence detection from video streams.

Tables 5 and 6 vividly illustrate that MobileNet V2 excels in violence detection, showcasing notable performance. While certain models, such as Resnet with LSTM and VGG 19 with LSTM, exhibit slightly superior performance across both scenarios, MobileNet V2 remains a standout performer. Particularly noteworthy is MobileNet V2's advantage as a lighter model, rendering it exceptionally well-suited for mobile devices. The pragmatic choice of integrating MobileNet V2 into the CMS not only ensures commendable performance but also underscores efficiency. This strategic utilization aligns with the imperative of optimizing CMS functionality, affirming MobileNet V2 as a robust choice for achieving enhanced and efficient violence detection capabilities.

Fig. 8 depicts some real-time violence detection scenarios based on human interaction and body postures. During any interaction between two people, the system detects abnormal behavior.

Fig. 8a depicts two people in a neutral position. As a result, the system detects no violence in this image. As the incident progresses, a conflict develops between the two people, and they engage in a fistfight. Throughout the images, different postures can be seen on the people involved. For example, in Fig. 8c one of them is being punched. Fig. 8d and Fig. 8e show different postures such as slapping, pushing, and kicking. The system was able to detect violence in every different situation.

4.4.2. Weapon detection

The main tool for weapon detection is the YOLOv5 object detection model. As a result, the model's performance is evaluated using the aforementioned model's parameter. Prior to model prediction, the image data was labeled using YAML class files. Sample weapon detection dataset with the label is shown in Fig. 9.

The precision and recall curve depicts the model's overall performance at various confidence levels. The system is observed to detect weapons with average confidence, whereas the detected weapons have a much higher confidence level. Fig. 10a shows the R-curve and Fig. 10b shows the P-curve for the weapon detection model.

Based on precision and recall, the F1 score can assist in determining the optimal confidence level at which the model performs best. Fig. 11a shows that all classes have the highest F1 score of 0.78 at the 0.276 confidence level. Where the F1 score is 0.75, the optimal confidence level could be around 0.4. The curve then begins to steeply decline.

The YOLOv5 model used in CMS can detect 5 classes of weapons with a collective F1 score of 86%. In the paper Sumi and Dey (2023) authors Lucy Sumi and Shouvik Dey describe a weapon detection YOLOv5 model utilizing data augmentation to generate and compare different results. For this purpose, three different kinds of datasets were used. Synthetic, mock attacks, and general images. Synthetic images are created using the Unity engine. Mock attacks are played out acts of violence and general images are the images of weapons taken from the internet or other sources. The dataset that was used to train the YOLOv5 for CMS can be compared to the general image data set used by the authors. The authors trained and tested the YOLOv5 model on a general dataset and achieved an F1 score of 78.2%. After data augmentation, the F1 score slightly improved to 79.6%. The YOLOv5 model used in CMS has a bit higher accuracy and achieved an F1 score of 86%. The mock dataset used by the author can be compared to CMS' real-time weapon detection. The model trained with a mock dataset performs with an F1 score of 64.5%. During the live mock performance, CMS could detect every weapon with good accuracy. In the paper Jain et al. (2020) the authors propose a weapon detection system that can detect weapons of different types and models using Faster RCNN and Single Shot Detection Algorithm. The Faster RCNN achieved an average accuracy of 84.6% and SSD achieved an accuracy of 73.8%.

For real-world evaluation, the model is tested while holding a replica weapon. The model correctly identified the weapon replica. Fig. 12 contains some example images. This information was obtained from a live video feed.

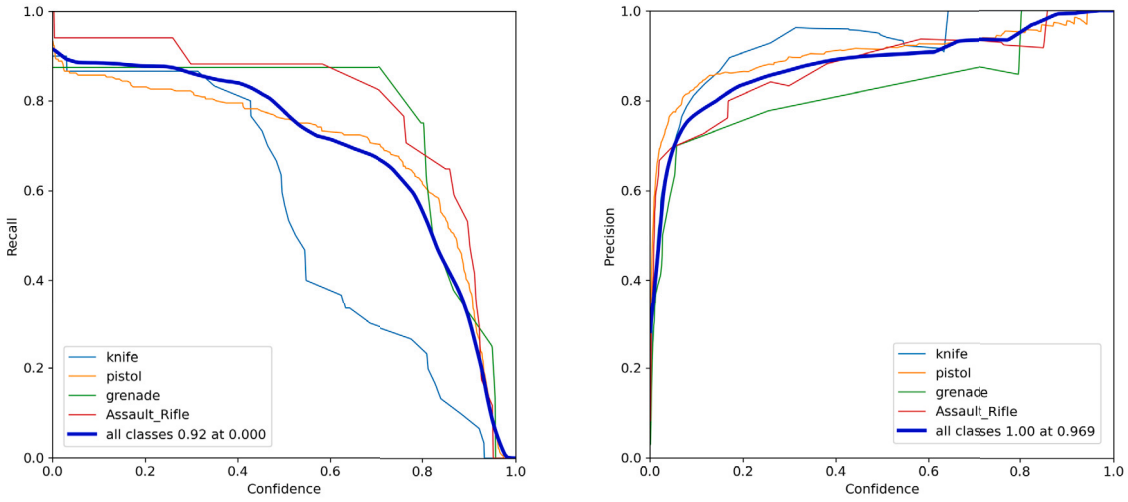
The system is designed to handle a diverse range of extracted image frames from videos, encompassing various shapes, forms, and modes. It is imperative that the system possesses the capability to detect criminal activities from varying angles and rotations. Specifically, the weapon detection module within the system is trained meticulously to identify weapons regardless of their orientation and from optimal distances. As depicted in Fig. 12., the weapon detection module exhibits remarkable performance. Its competence extends to detecting weapons effectively, irrespective of the angle or rotation at which they appear. Rigorous testing involving scenarios featuring individuals holding weapons was conducted. These scenarios involved rotating the image frames to simulate different perspectives. Remarkably, the weapon detection module integrated into the Content Monitoring System (CMS) showcased its prowess by consistently detecting weapons from all angles and orientations. The efficacy of the module was confirmed through a diverse set of images featuring various weapon types, such as knives, guns, and assault rifles. The module's discerning capability extended to accurately categorizing these weapon types from the images. This outstanding performance renders the weapon detection module perfectly suited for the task of Crime Monitoring, solidifying its contribution to the system's capabilities.

4.4.3. Face detection and recognition

Local Binary Pattern Histogram (LBPH) is used to create the face detection and identification model. The model's ability to recognize a person's face can be used to assess their performance. A dataset containing the team members' face images was provided for training and testing purposes. The model can recognize facial features from vari-

Fig. 8. Real-time scenarios of violence detection. The face has been blurred for privacy issues.

Fig. 9. Weapon detection dataset.



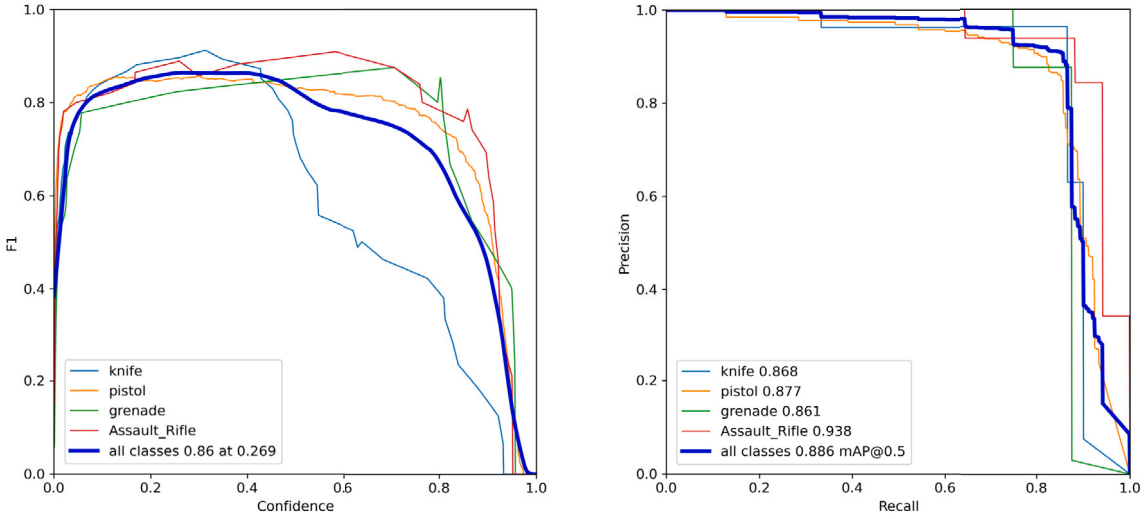
(a) R-curve for Weapon detection.

(b) P-curve for Weapon detection.

Fig. 10. R-curve and P-curve for Weapon detection.

ous angles. The input is a live video stream taken frame by frame. The person in front of the camera must be detected and identified by the system.

To further evaluate the performance of the CMS face recognition model, it can be compared with similar works. In Wang et al. (2017), authors Wang et al. describe a face recognition system in a real-world surveillance system using deep learning methods. The authors used a fine-tuned VGG face model with an overall accuracy of 92%. The VGG face model used by authors Ghazi et al. in the paper Ghazi and Ekenel (2016) achieves an accuracy of 98.9%. The state-of-the-art model FaceNet proposed by authors Schroff et al. in Schroff et al. (2015) achieves an accuracy of 99.63%. The proposed Face recognition method in CMS uses haar cascade face detection and LBPH face recognition to achieve an accuracy of 97%.



(a) F1-curve for Weapon detection.

(b) PR-curve for Weapon detection.

Fig. 11. F1-curve and PR-curve for Weapon detection.

Table 7
Comparison Table performance summary.

	F1	Accuracy		
		Gun	Rifle	Average
Proposed YOLOv5	0.86	0.877	0.938	0.886
Lucy-Shouvik YOLOv5 Sumi and Dey (2023)	0.795			0.82
H.Jain- Faster RCNN Jain et al. (2020)		0.74	0.94	0.846
H.Jain-SSD Jain et al. (2020)		0.66	0.80	0.738

Table 8
Face recognition performance comparison.

Models	Performance
VGG Face (Real time) Wang et al. (2017)	92%
VGG Face Ghazi and Ekenel (2016)	98.9%
FaceNet Schroff et al. (2015)	99.63%
Proposed method (Real-time)	97%

Upon careful examination of Tables 7 and 8, it becomes apparent that the innovative CMS method exhibits commendable performance in the realm of face recognition tasks. Trained on a custom dataset, the model undergoes real-time testing, providing valuable insights into its practical efficacy. A benchmark comparison reveals that while the state-of-the-art FaceNet model achieves an impressive accuracy of 99.63%, the proposed CMS method holds its own with a robust accuracy of 97%. This compelling performance underscores the model's suitability for the designated task. To further enhance its utility, the model undergoes meticulous tuning, ensuring swift and efficient face detection and recognition within a constrained time window.

Fig. 13 shows a set of images used to detect faces from various angles. The front profile of the face is detected in Fig. 13a. This angle has a confidence level of 69.70%. The side profiles of the face are used to test the system in Fig. 13c and 13c, and it can detect with the confidence of 87.45% from the right and 81.55% from the left. When the person in Fig. 13d is looking upward, the system can detect from below the chin with a confidence of 84.77%.

According to the results, the system had the highest degree of accuracy when detecting faces at a distance. When detecting from a close distance, it had average accuracy. The model's performance is clearly influenced by the distance between the subject and the camera.

4.4.4. Alerts

Several models, including weapon detection, facial recognition, and violence detection, are used by the system to detect crime incidents. The system's performance is heavily reliant on the accuracy and efficiency of the individual models. Based on the previous evaluation, the individual models performed satisfactorily, with high accuracy and low time complexity. When the models were combined, their performance did not suffer, allowing the system to detect crime incidents in real-time.

However, other factors, such as the quality of the security camera that captures live video, may have an impact on the overall performance of the system. Even a minor lag in video capture could jeopardize the system's ability to detect crime in real time. Furthermore, if the video quality is poor, extracting features from image frames may be difficult, resulting in a decrease in system performance.

As a result, while the system has the potential to perform well under appropriate conditions, several other factors may have an impact on its performance. To ensure the system's optimal performance, it is critical to consider these external factors when implementing it.

Table 9 illustrates how CMS can generate various alert levels based on the observed incident. The system observes two individuals in a neutral state in the first scene and does not generate an alert. The CMS observes the two individuals engaged in hand-to-hand combat in the second scene, but no weapon is involved. In such a case, the system sends a 'Caution' alert to the authorities. The CMS notices one of the individuals threatening the other with a weapon in the third, fourth, and fifth scenes. The CMS generates a 'Danger' alert based on the severity of the situation to bring the situation to the attention of the appropriate authorities. The CMS notices an individual armed with a weapon in the final scene, but no one else is in the area. In this case, the CMS issues a 'Warning' alert to ensure that the appropriate authorities are notified and can take appropriate action.

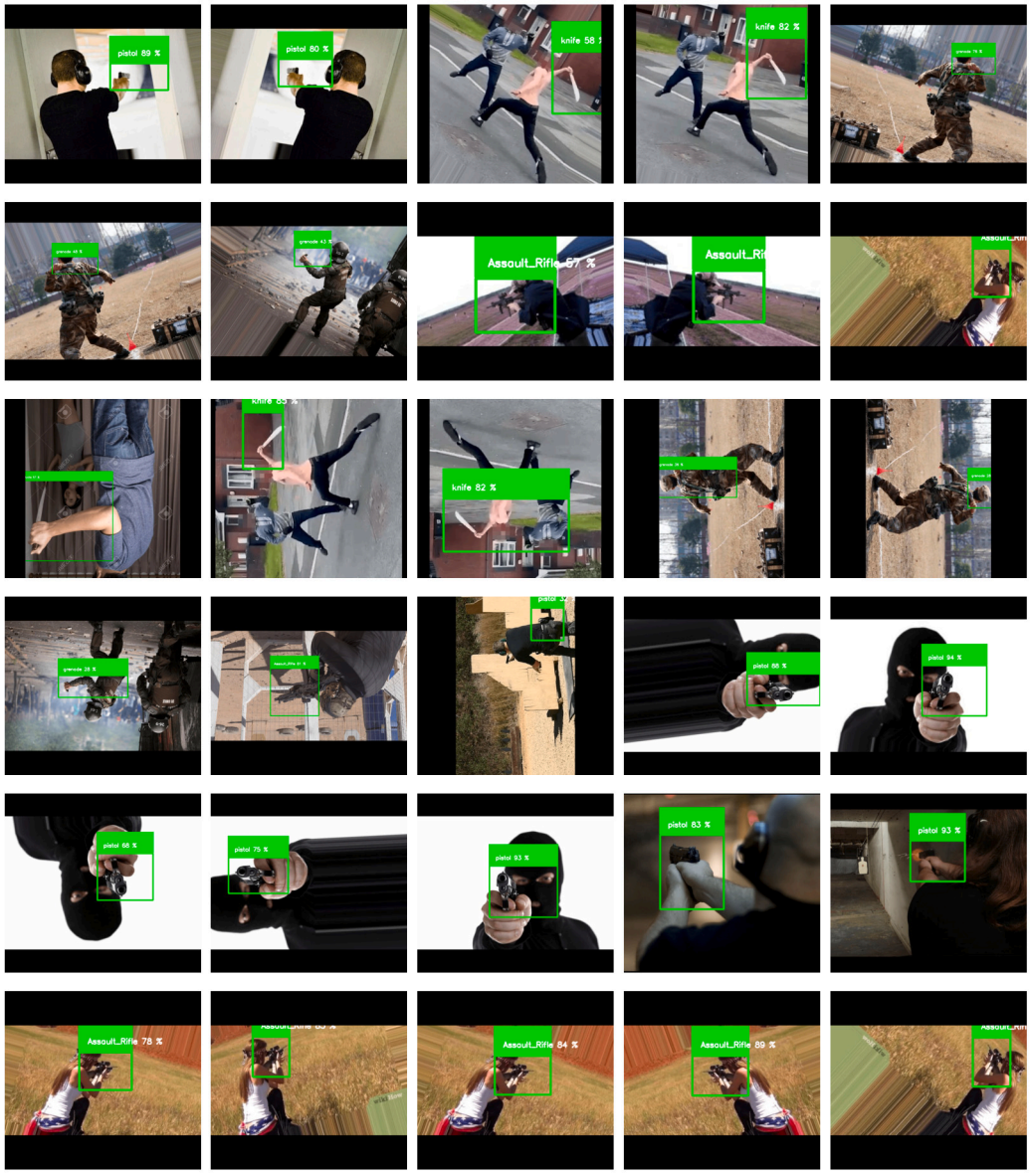
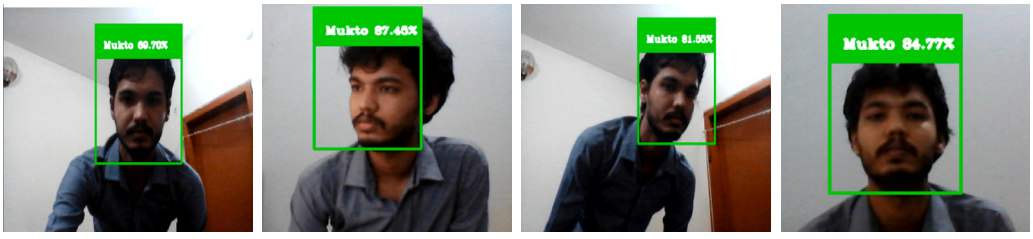


Fig. 12. Real-time scenarios of weapon detection.



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

Fig. 13. Real-time scenarios of face detection.

5. Conclusion and future work

We all want to live in a society where we can feel safe and secure without constantly being on high alert. Various crime prevention measures were implemented with varying degrees of success in order to make this ideal society a reality. An automated crime detection system not only helps to prevent crimes from occurring, but it also has some psychological effects. The paper's proposed system detects and identifies crime incidents by combining multiple image processing and deep learning methods. To achieve the best results, only the best methods and algorithms were used. The system was tested for compatibility using a variety of methods and models. While training and testing the system, a combination of methods and models such as Local Binary Pattern Histogram (LBPH), YOLOv5, and MobileNet generated the most effective results. In its current state, the system can detect crimes from live camera feeds using methods such as face detection, violence detection, and

Table 9
CMS alert levels Results.

Scene	WD	IPETO	ABD	Alert
	False	False	False	-
	False	False	True	Observe
	True	True	True	Danger
	True	True	False	Danger
	True	True	False	Danger
	True	False	False	Warning

weapon detection. Multiple methods, such as weapon detection and face detection, can be used to identify a crime incident. When there are no weapons present, violence detection can be used. The CMS employs a variety of methods to detect crime as quickly as possible. The face detection method can be used to identify a law enforcement officer. Despite the fact that the system was designed to provide real-time security in any situation at any time, it does have some limitations. The system may be unable to detect crime in areas with poor lighting because detecting an object becomes difficult. The features and structure of an object cannot be properly defined without proper illumination, or even with inadequate lighting. As a result, identifying an object becomes more difficult. In the case of night vision and thermal imaging cameras, the feature of an object becomes hazy, and the system may be unable to detect objects because a different dataset is required. As a result, the system may struggle to detect crime from low-quality images or video frames. When a person carries a weapon, he or she does not keep it in his or her hand. They are usually handcuffed until they commit a crime. At this time, the system may be unable to detect holstered weapons or weapons that are completely concealed. Because the system identifies crimes by running multiple processes at the same time, it may perform poorly on low-spec hardware, and the time complexity may increase.

In our upcoming endeavors, we aim to enhance crime detection capabilities in low-light environments through the integration of advanced night vision image processing. Additionally, the incorporation of X-ray imaging into our system is envisioned to elevate our ability to identify concealed weapons, whether they are hidden in undergarments or within bags. A paramount focus of our work involves optimizing the system for heightened efficiency. The refinement of our crime detection procedures will leverage a sophisticated combination of video and sound analysis techniques. This integration not only ensures a more comprehensive understanding of the environment but also contributes to the precision of our crime detection mechanisms. In the event of a detected crime, we plan to introduce features that enhance the system's responsiveness. This includes the implementation of mobile notifications and real-time alerts, providing timely information to relevant stakeholders. To further elevate the efficiency of our model, we are embracing edge computing technology. By introducing edge computing, our surveillance cameras, equipped with the necessary CPU and GPU capabilities and integrated with a robust CMS system, will possess the capability to autonomously detect and respond to criminal activities without transmitting the entire video feed to a centralized server. This decentralized approach not only reduces latency but also optimizes bandwidth usage, contributing to a more agile and responsive crime detection system. As part of our exploration into cutting-edge technologies, we are conducting tests with autonomous surveillance drones. These drones, integrated with our CMS, are poised to capture video frames from larger areas, extending the reach and capacity of our crime detection system. This integration showcases our commitment to embracing innovative solutions that leverage both ground-based and aerial perspectives for a more comprehensive surveillance approach. In summary, our future initiatives are centered around advanced image processing, efficient crime detection procedures, and the strategic integration of edge computing and drone technologies. These endeavors collectively aim to establish a robust and responsive system that excels in crime detection across diverse and challenging environments.

CRediT authorship contribution statement

Md. Muktadir Mukto: Conceptualization, Methodology, Software and Original draft preparation, Visualization, Writing – Editing and Validation. Mahamudul Hasan: Conceptualization, Methodology, Original draft preparation, Writing – Editing and Validation, Final Review. Md. Maiyaz Al Mahmud, Ikramul Haque, Md. Ahsan Ahmed: Dataset Preparation, Result Interpretation, Revision. The rest of the authors contributed to supervising the project and revising the draft. All authors have read and agreed to the published version of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Rai, M., Husain, A. A., Maity, T., & Yadav, R. K. (2018). Advance intelligent video surveillance system (AIVSS): A future aspect. *Intechopen*. https://doi.org/10.5772/intechopen.76444.

Sung, C.-S., & Park, J. (2021). Design of an intelligent video surveillance system for crime prevention: Applying deep learning technology. *Multimedia Tools and Applications, 80*. https://doi.org/10.1007/s11042-021-10809-z.

Hajri, F., & Fradi, H. (2022). Vision transformers for road accident detection from dashboard cameras. In *2022 18th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1–8).

Rasheed, N., Khan, S. A., & Khalid, A. (2014). Tracking and abnormal behavior detection in video surveillance using optical flow and neural networks. In *2014 28th international conference on advanced information networking and applications workshops* (pp. 61–66).

Wei, H., Laszewski, M., & Kehtarnavaz, N. (2018). Deep learning-based person detection and classification for far field video surveillance. In *2018 IEEE 13th Dallas circuits and systems conference (DCAS)* (pp. 1–4).

Zhou, J. T., Du, J., Zhu, H., Peng, X., Liu, Y., & Goh, R. S. M. (2019). Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, *14*(10), 2537–2550. https://doi.org/10.1109/TIFS.2019.2900907.

Xu, J. (2021). A deep learning approach to building an intelligent video surveillance system. *Multimedia Tools and Applications*, *80*, 1–21. https://doi.org/10.1007/s11042-020-09964-6.

Motiian, S., Siyahjani, F., Almohsen, R., & Doretto, G. (2017). Online human interaction detection and recognition with multiple cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(3), 649–663. https://doi.org/10.1109/TCSVT.2016.2606998.

Navalgund, U. V., & P., K. (2018). Crime intention detection system using deep learning. In *2018 international conference on circuits and systems in digital enterprise technology (ICCSDET)* (pp. 1–6).

Verma, G. K., & Dhillon, A. (2017). A handheld gun detection using faster r-cnn deep learning. In *Proceedings of the 7th international conference on computer and communication technology* (pp. 84–88). New York, NY, USA: Association for Computing Machinery.

Mukto, M. M., Al Mahmud, M. M., Haque, I., Imam, O. T., Reza, A. W., & Arefin, M. S. (2022). Developing a tool to classify lethal weapons by analyzing images. In J. I.-Z. Chen, J. M. R. S. Tavares, & F. Shi (Eds.), *Third international conference on image processing and capsule networks* (pp. 229–242). Cham: Springer International Publishing.

Buckchash, H., & Raman, B. (2017). A robust object detector: Application to detection of visual knives. In *2017 IEEE international conference on multimedia & expo workshops (ICMEW)* (pp. 633–638).

Alaqil, R. M., Alsuhaibani, J. A., Alhumaidi, B. A., Alnasser, R. A., Alotaibi, R. D., & Benhidour, H. (2020). Automatic gun detection from images using faster r-cnn. In *2020 first international conference of smart systems and emerging technologies (SMARTTECH)* (pp. 149–154).

Grega, M., Łach, S., & Sieradzki, R. (2013). Automated recognition of firearms in surveillance video. In *2013 IEEE international multi-disciplinary conference on cognitive methods in situation awareness and decision support (CogSIMA)* (pp. 45–50).

Harikrishnan, J., Sudarsan, A., Sadashiv, A., & Ajai, R. A. (2019). Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision. In *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)* (pp. 1–5).

Ben Ayed, M., Elkosantini, S., Alshaya, S. A., & Abid, M. (2019). Suspicious behavior recognition based on face features. *IEEE Access*, *7*, 149952–149958. https://doi.org/10.1109/ACCESS.2019.2947338.

Takai, M. (2010). Detection of suspicious activity and estimate of risk from human behavior shot by surveillance camera. In *2010 second world congress on nature and biologically inspired computing (NaBIC)* (pp. 298–304).

Chumuang, N., Ketcham, M., & Yingthawornsuk, T. (2018). CCTV based surveillance system for railway station security. In *2018 international conference on digital arts media and technology (ICDAMT)* (pp. 7–12).

Wang, J., & Xia, L. (2019). Abnormal behavior detection in videos using deep learning. *Cluster Computing*, *22*. https://doi.org/10.1007/s10586-018-2114-2.

Benito-Picazo, J., Domínguez, E., Palomo, E., & López-Rubio, E. (2020). Deep learning-based video surveillance system managed by low cost hardware and panoramic cameras. *Integrated Computer-Aided Engineering*, *27*, 1–15. https://doi.org/10.3233/ICA-200632.

Ramzan, M., Abid, A., Khan, H. U., Awan, S. M., Ismail, A., Ahmed, M., Ilyas, M., & Mahmood, A. (2019). A review on state-of-the-art violence detection techniques. *IEEE Access*, *7*, 107560–107575. https://doi.org/10.1109/ACCESS.2019.2932114.

Gong, A., Chen, C., & Peng, M. (2019). Human interaction recognition based on deep learning and hmm. *IEEE Access*, *7*, 161123–161130. https://doi.org/10.1109/ACCESS.2019.2951937.

Sumon, S. A., Goni, R., Hashem, N. B., Shahria, T., & Rahman, R. M. (2020). Violence detection by pretrained modules with different deep learning approaches. *Vietnam Journal of Computer Science*, *07*(01), 19–40. https://doi.org/10.1142/S2196888820500013.

Sumi, L., & Dey, S. (2023). Yolov5-based weapon detection systems with data augmentation. *International Journal of Computers & Applications*, *45*(4), 288–296. https://doi.org/10.1080/1206212X.2023.2182966.

Jain, H., Vikram Mohana, A., Kashyap, A., & Jain, A. (2020). Weapon detection using artificial intelligence and deep learning for security applications. In *2020 international conference on electronics and sustainable communication systems (ICESC)* (pp. 193–198).

Wang, Y., Bao, T., Ding, C., & Zhu, M. (2017). Face recognition in real-world surveillance videos with deep learning method. In *2017 2nd international conference on image, vision and computing (ICIVC)* (pp. 239–243).

Ghazi, M. M., & Ekenel, H. K. (2016). A comprehensive analysis of deep learning based representation for face recognition. In *2016 IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 102–109).

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 815–823).