IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING APPROACH FOR DETECTION OF FUGITIVES

[1]Dr.Jayavrinda Vrindavanam, [2]Dr.Raghunandan Srinath, [3]Mrs. M. Lakshmi, [4]Srishti Singh, [5]Tanisha Banerjee, [6]Tripti

[1]Associate Professor, [2]Principal Member of Technical Staff, Graylinx Pvt Ltd, Bangalore, [3]Associate Professor, [4]Student, [5]Student, [6]Student
[1]Department of Electronics and Communication Engineering, [3]Department of Information Science and Technology,
Nitte Meenakshi Institute of Technology, Bangalore, India

Abstract: Surveillance systems play a crucial role in maintaining security and order in public places. Despite CCTV cameras installed at most of the locations, crimes that are captured in CCTVs not reported in an effective way. In this context, there has been an enhanced attention for increased surveillance in order to monitor crowd behaviour and to nab fugitives, suspects and criminals, especially in bus stations, railway stations and airports and other crowded locations. CCTV, in general, requires human supervision which may lead to missing some important crime events due to human error. The proposed project aims to help automate the process of monitoring the CCTV footage and supports in proactively detecting suspicious behaviour in real-time videos with alerts sent to higher authorities. The alert is generated and forwarded to designated authorities if suspicious events occur, which will help in preventing crimes, assisting in crime investigation, identification of suspects and also supports in automating some of the groundwork that is manual and time-consuming. Towards this, the paper attempts to incorporate two independently working machine learning models into one working model. To get the best results, transfer learning has been used for enhanced performance of each model and give better and faster results. The video feed from the CCTV will be taken as the input in our system and image frames will be extracted and fed into the two sub-systems of the project. For criminal/suspects face recognition, transfer learning on VGG16 model is used, for weapon detection, darknet framework is to be applied with YOLOv4 algorithm to get the desired object detection. These machine learning systems have been designed to work in parallel which in turn ensures faster output.

***Index Terms*–CCTV surveillance, machine learning, transfer learning, face recognition, VGG16, YOLOv4.**

I. INTRODUCTION

Closed-circuit television (CCTV) is the use of video cameras to transmit a signal to a selected destination control room, on a limited set of monitors, where people watch all the feeds for monitoring. With the increased use of machine learning supported by ever-increasing computing power, these automated surveillance systems can conveniently be an efficient tool in aiding enforcement agencies as machine learning has been extensively used for detection, recognition, observation, tracking and helping in the recognition of the suspicious activity within the surveillance environments. At present, there is a lot of human interventions required to detect criminal behaviour in surveillance videos, which is time-consuming and prone to errors. Observing videos of large volume at all the points in time by the surveillance staff may lead to omissions in observation and highlighting of suspicious scenes and tasks can get complicated with multiple footages to be observed. The paper proposes to address this issue by developing an automatic surveillance system that can detect criminal behaviour with less computational time.

The proposed system has been designed to solve problems occurring at Public Places where CCTV cameras are present but not properly monitored. The proposed system aims to integrate the two machine learning models and make them work seamlessly. First is the criminal face detection algorithm that will check for any known or wanted criminals with the support of transfer learning from the VGG16 model. This will give an accurate prediction for the classification. For our project, a custom dataset of criminals is created by using the faces of friends and family and used to train the model (in the absence of limitations in accessing such database from law enforcement agencies). The second model is Weapon detection, which is nothing but specific object detection. This model aims to detect the weapons in the hands of persons which will be useful to proactively detect the crimes before it occurs with the use of Deep Learning YOLO algorithm. Deep Learning YOLO concepts have been generally preferred on account of their merits in saving time, memory and resources like CPU and Processers. YOLO framework models can also provide more accurate results compared to designing a machine learning model explicitly from scratch, which is also referred to as Transfer Learning. For this project, YOLO v4 darknet models are trained on a custom dataset of guns, generated using Roboflow, which is a developer tool used to build computer vision models faster and more accurately for better results.

II. A REVIEW OF RELATED WORKS

Keeping in view the objective of bringing together different techniques that can support surveillance under a machine learning model, we observed that, the works in this direction are fairly spread out. Accordingly, plenty of papers with a slightly different focus of study were considered for this work. In this section, we have divided the literature review into two parts, face Recognition and object detection.

2.1. Face Recognition

The face recognition methodology has been a well-established approach and a large number of studies are already in place. Face recognition problem can be formulated as a given still or an image frame from a video of a scene, in which one needs to identify one or more persons in the scene by comparing with faces stored in a database. Face detection is the first stage of any face recognition system. The face detector needs to be fast as well as accurate in order to build an efficient end-to-end face recognition model. Hence, we require a robust face detector that provides the output in a single pass of the network. In order to detect faces at different scales, Haar Cascade based face detection is found to be the most trusted face detector and it can also support [1] in identifying more than one face in a single frame by pre-processing the image and feature extraction using Haar Cascade Classifier Algorithm, the classifier is the Viola-Jones method by selecting few significant features using AdaBoost. Sirovich and Kirby [2] developed a face recognition system using the Eigenfaces approach that was later enhanced by Turk and Pentland [3]. This has offered a breakthrough in the field of face recognition system and formed the basis of the Face Recognition Algorithm. A nearly real-time computer system was developed, that could locate a person's head and then recognize their face by comparing a characteristic of the face to those that are in the dataset. In later papers, a face recognition system was proposed that used PCA principal component analysis (PCA) dimensionality reduction and neural networks for classification. With successive improvements, the recent paper [4] introduced face identification using keyframes concepts with the Eigenface and PCA method and finally a comparison based on Euclidian distance with Eigenfaces and restored Eigenfaces with the dataset and predicting the results.

Transfer learning is the process of transferring previously learned knowledge from an established working model, to form a new task-specific model that may be related to the original task of the source model. This is done to avoid designing a model from the scratch and ensures efficient training. To train a face recognition Convolutional Neural Network (CNN) model from scratch, both a big dataset and powerful computing resources are necessary requirements, which hinder many aspirants in the field. Transferring the weights of an already learned face model to adjust according to our needs is more suitable for accuracy as well as speed and also for saving computational cost. Parkhi et al. [5] in their paper, the CNN model was trained on a large-scale dataset containing 2.6 million images of 2,622 people and achieved commendable results. It is a publicly available model known as VGG_FACE, and it also supports Caffe, Torch, and MatConvNet. In another work, [6] the authors suggested that higher layer features are more global while lower layer ones are also more local and that the higher layers have global attributes that help in better inter-class differentiation, and more easily trainable. This is the reason why changes are made to the last layers while freezing the initial layers. Because of the already learned source model, trained on a large dataset, transfer learning allows the target model to achieve almost negligible error rates even with very limited training samples.

2.2. Object Detection

Object detection is the process of locating and classifying existing objects in an image and enclosing them within labelled rectangular bounding boxes. Generally, there are two basic frameworks of object detection. Firstly, the traditional object detection method where region proposals are generated at first and then each proposal is classified to predict the existing object. The second framework is where object detection is regarded as a regression or classification problem, and then a unified framework is adopted to predict the different categories of objects and their respective locations. Due to the strong learning ability of the deep neural network, the problem with any deep learning model is that the accuracy will be low for limited sample conditions, often resulting in an overfitted model with drastically reduced performance. In a study, CNNs were used in hybrid models where both feature extraction and classification were performed by the same model [7]. Different layers like convolution, pooling and fully connected layers were used to form a convolutional neural network. The convolution layers were associated with a number of filters, whose weights were randomly initialized in the beginning, and used for convolution across the height and width of the input feature. The weights in the filters were then updated with each iteration during the training process. The pooling layer performs dimensionality reduction of input samples for easier computation for subsequent layers[8].

The fully connected layers that are referred to as the top or higher layers are used for the final classification. It is because of this reason, that in most transfer learning models, the fully connected layers are removed while importing from the original model[9]. You only look once (YOLO)introduced in 2016 by Redmon et al. [10] brought a major revolution in the solution of the object detection problem because they viewed object detection as a regression problem instead of a classification problem. The implemented method was to divide an image into an S x S grid (a 7 x 7 grid in the case of the paper). For each grid, an (x, y) input labelled data was generated and fed to a CNN. Here, the information about the class of the object and the bounding box enclosing it is contained in the vector labelled 'y'. Since all the grids are given as input to a convolutional implementation of sliding window detection, so the yolo model can detect all the objects in the grids, in one single pass of the image. Over the years, with the introduction of newer versions, YOLOv2[11] improved the use of anchor boxes to detect multiple objects in one grid. In YOLOv3[12], a significant change made was in the use of k-means clustering for bounding boxes instead of the traditional IOU metric function. YOLOv4[13] was proven to be faster and improved the FPS of YOLOv3 by 12%.

In this recent paper[14], tracking of an object along with a person was also proposed which helps in better analysis of the situation. This paper discusses tracking of a piece of unmanned luggage and also the person as an object and sending an alert to higher authorities in an airport environment. This was done using the FRCNN inception v2 framework and multi-camera system,

III. PROPOSED METHODOLOGY

The proposed system aims to design a Machine Learning(ML) algorithm that works for both the sub-systems and gives accurate results to predicts the scenario. Keeping in mind the accuracy of our final model, transfer learning is used in each of the sub-systems. In the implementation, image frames are to be extracted from the CCTV video footage. These extracted frames are then fed as input into the two sub-designs of the project, each of which checks for respective components in the input frames:

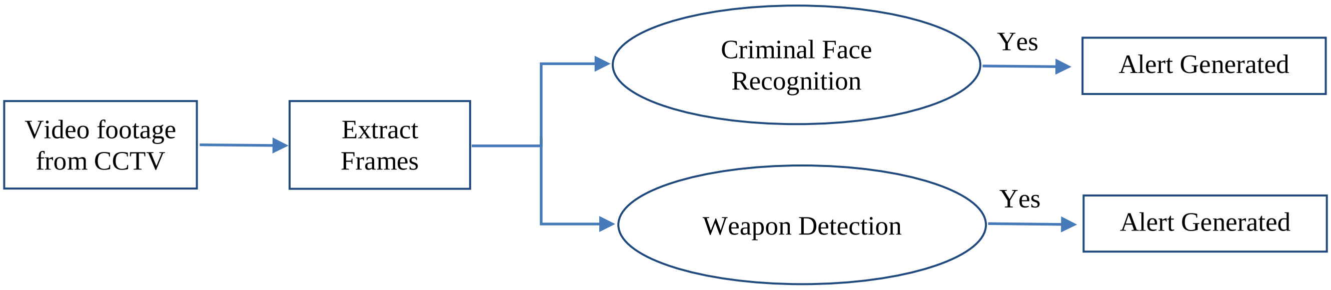


Fig 3.1: Process flow structure of the Project

For this process to perform, the extracted frames are copied as a different variable. The original frame is sent to the criminal face recognition system, and the copied frame is sent to the weapon detection system. Since each model employs a different way of processing the frames, the resizing of the frames is done accordingly and fed to the respective detection system. When the respective output is obtained from the two modules, the output display frame from the second module is resized to the shape of the output frame of the first module. Then both these frames are then concatenated together to monitor different targets in the same scene parallelly. Each of the modules is associated with a third-party software, Twilio, with the help of which alert can be sent to specified authorities in case if their respective targets are detected. This is not dependent on whether the other module has detected its target or not. When the criminal face recognition module detects a criminal/suspect in the frames, an alert will be sent irrespective of the weapon detection module.

3.1 Criminal Face Recognition

3.1.1 Dataset Preparation:

For the custom dataset, pictures of family and friends have been used. There are a total of 800 images that are considered. These images are then increased by means of data augmentation. ImageDataGenerator is used to rescale each pixel value of each image to a range of [0,1]. The rotation range is kept as 20 degrees, while the height and width shift range is kept as 0.2 to take only the face region into consideration, the horizontal flip is set to true.

3.1.2 Design of the model:

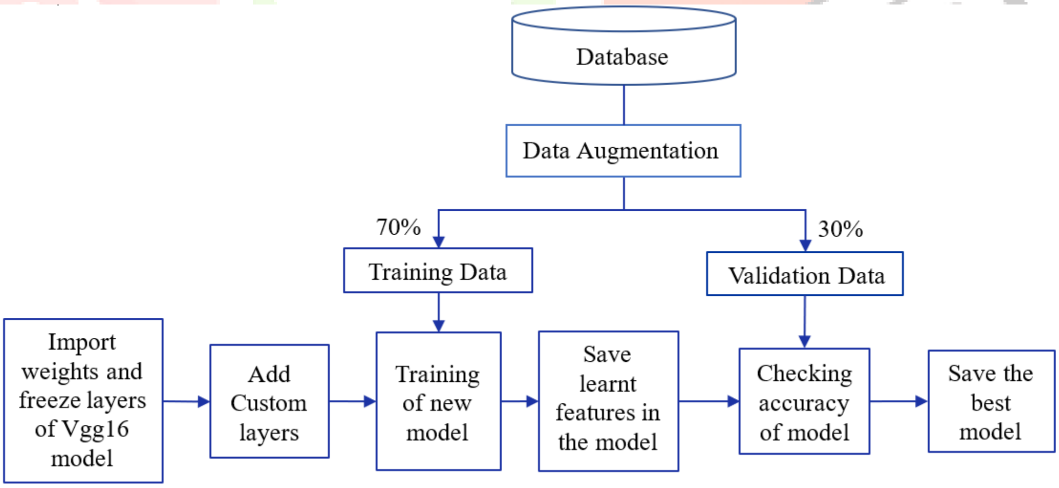


Fig 3.2: Flow Diagram of the Criminal face identification model

The widely popular VGG16 is used for developing the face recognition model. VGG16 is different from the original VGG_FACE model that was released by Oxford University, concerning the fact that it was modified, to be trained on the ImageNet dataset, and to give an output of 1000 classes. Its parameters have been changed from the original VGG_FACE to 16 trainable layers, thus earning the name VGG16. Originally, VGG_FACE was developed as a deep CNN model for face recognition trained on 2.6M images of 2,622 faces. It consists of 40 layers, including an input layer (known as layer 0) and the output layer (layer 39) with SoftMax activation function. The model has 3 fully connected layers (layer 32, 35, 38), with alternating convolution, ReLU, maxpool, drop layers for the remaining model.

We import the weights of VGG16 model and copy the lower layers of the existing model to form the first 18 layers of our model and call these layers as the source model. The aim is to fuse these imported layers with our custom layers and train them on our custom dataset to obtain the final criminal face recognition model. Here, the imported layers from the source model are frozen, so as not to disturb their already learnt weights, during training the model. This is done so that these weights and initial layers are preserved as it is, which will actually help us in the initial training of the custom layers that we add for the final model.

After we copy the lower layers of source model VGG16 we freeze them. Then we add our custom layers to it. The layers added are GlobalAveragePooling2D, and four dense layers. Global average pooling layer reduces the dimensionality of the feature maps output by the last convolutional layer of our frozen model. It is used instead of flatten layer, to give us a two-dimensional matrix, from the previous four-dimensional matrix of the last layer. The next four dense layers are to further reduce the output shape to finally give us the output layer of two classes: Criminal or Innocent.

Here feature extraction of the images in the custom layer is done using the frozen layers of the source model. The extracted features are then used to train the custom higher layers that were added. Thus, the frozen lower layers imported from the source model combined with the custom layers added on top of them and trained with custom data forms the architecture of our criminal face recognition model. For the feature extraction from an image, VGG16 has an input dimension of 244x244x3, where the picture is resized to 244 x 244, and the number of channels is set to 3. For the unbiased training of the model, images of both labels are used for training, that is, criminal as well as innocent labelled images. The optimizer used for the training is Adam, and the early stopping function is employed to avoid overfitting.

3.2 Weapon Detection

3.2.1 Dataset Preparation

In order to train on the custom dataset of weapons, namely guns, with YOLOv4, it is of utmost importance that the dataset is labelled for aiding the training. There are many open-source GUI tools to help us easily generate label file from the image. One such tool used is Roboflow, a computer vision developer tool. Bounding boxes were created around the target object, manually for each of the 7000 images in the dataset, then the tool generated the label file automatically. Each image from the dataset of 7000 images is associated with a **.txt** file having the same name, which contains the object classes and their coordinate following this syntax: ***<object-class> <x_center> <y_center> <width> <height>***

Three such files were created: ***classes.names***, ***train.txt*** and ***test.txt***. Classes.names contains the label of the object to be detected, which in our case is a single class of guns. Train.txt and test.txt contains the training and testing data, respectively.

3.2.2 Design of Model

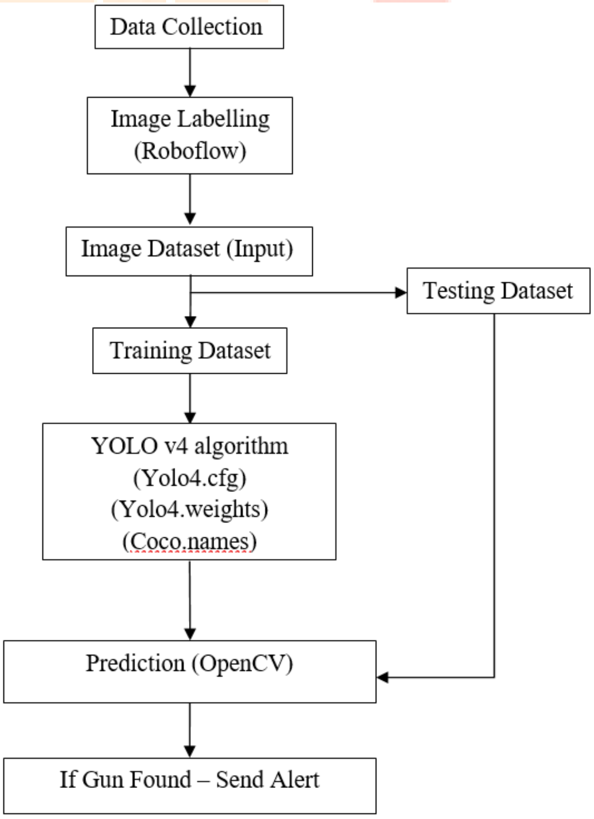

Data Collection
Image Labelling (Roboflow)
Image Dataset (Input)
Testing Dataset
Training Dataset
YOLO v4 algorithm (Yolo4.cfg) (Yolo4.weights) (Coco.names)
Prediction (OpenCV)
If Gun Found – Send Alert

Fig 3.3: Flow Diagram of the Weapon detection model

The model is trained with Darknet framework and YOLO v4 algorithm. Darknet framework is an easily available open-source neural network that is based on C and CUDA. Being based on C, the darknet framework has less complexity and is more computationally cost-effective thereby making it extremely efficient and fast in training a model. It also supports both CUI and GUI computation and is arguably one of the best neural networks for real-time custom object detection application.

For the building of the model, the darknet neural network is imported into our system. The configuration file of the original framework is re-written, with our custom model in consideration. The max batches set were 2000, while we worked with 64 batches, with 24 subdivisions. Here since we are training the model for only gun detection, so our classes file consists of only one class. Each of the training pictures is of the dimension 416 x 416, with 3 channels and the learning rate is set to 0.001.
Re-writing of the configuration file in the original network has certain changes in the three parts of yolo, namely backbone, neck and head, and the immediate convolution layer before each. Mainly here we set the number of classes to be detected using the custom detection model.

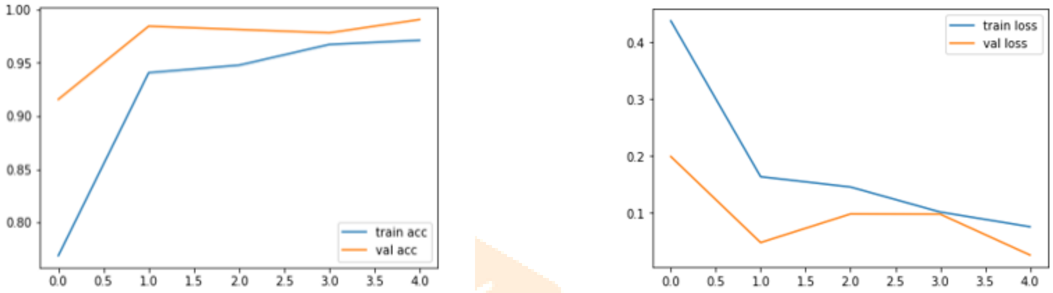
In this project, the number of classes is set to 1 because there is only one class of object: gun.
Then filter size of the convolutional layer before each yolo component can be chosen according to the formula,
filter = (number of classes + 5)*3
For the project, since the number of classes = 1, so we have, filter = 18

The data is split into training and validation sets and training of the model starts, with 2000 iterations. The IOU threshold is set to 50% and mAP is calculated at every interval of 500 iterations to save the best weights. After sufficient training and validation, the weights and configuration of the network are saved. The custom object detection network can now be loaded on our system and using the configuration and weights file from the training, object detection can be done on a video stream.

At the very end, if the targeted object, that is guns are detected in the frames, an alert is to be sent to a specified phone number of higher authorities, using third party software, Twilio.

IV. Result and Discussion

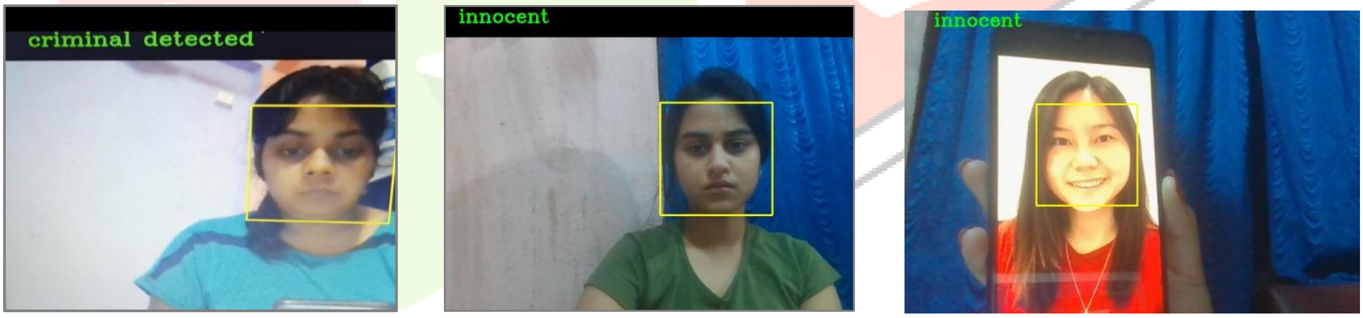
For both the models, the validation accuracy is used as the evaluation parameter to test the performance.
The Criminal face identification model is successfully trained on our custom dataset, with 10 epochs, employing an early stopping function. The early stopping function is to make sure the model is not overfitted, so the patience set for it was 3, which means that if the validation accuracy is not improved in the next 3 iterations, then the model stops training, and the best weights are retained.
From the training of the model, the validation accuracy was 98.44%, the training accuracy was 97.66%.



(a) (b)

Fig 4.1(a): Plot for the training and validation accuracy; (b): Plot for the training and validation loss

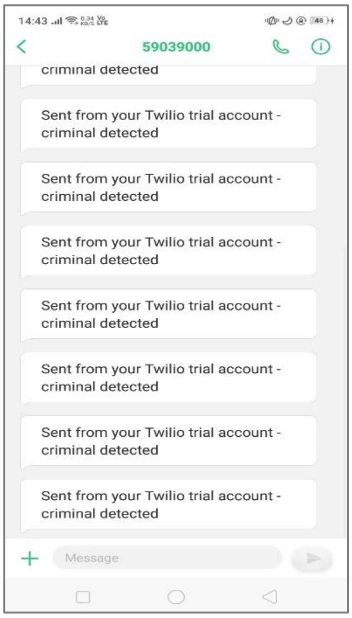
For the output of the Criminal face recognition model, a bounding box is created around the face of the criminal and an alert is instantly generated to concerned authorities.



(a) (b) (c)

Fig 4.2 (a): Successful identification of criminal; (b): Successful identification of innocent; (c): Faces that are not in the dataset are defaulted as innocent

From figure 4.2, we conclude that the model successfully identified the respective persons in the criminal and innocent labelled dataset. Also, if the face is not in the records they have defaulted as innocents as we can see in figure 4.2(c). The alert is sent for the criminal detection only and not for the innocent person as shown in figure 4.3



criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

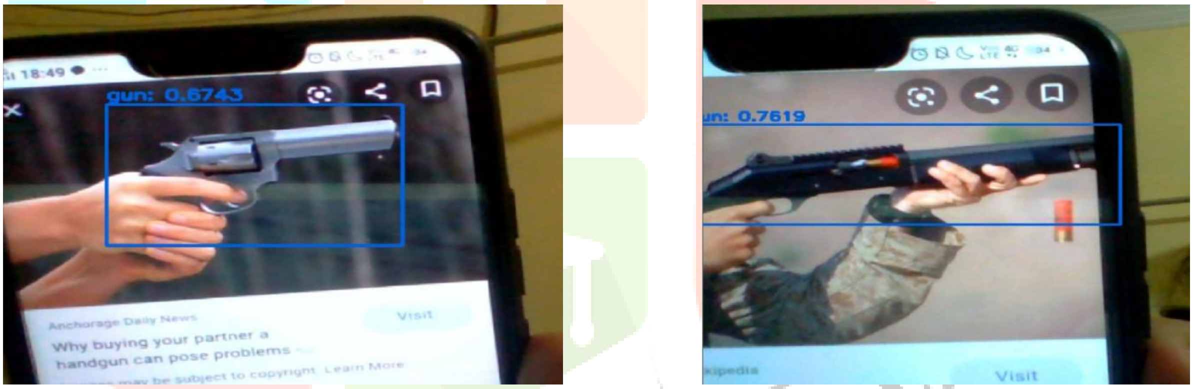
Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

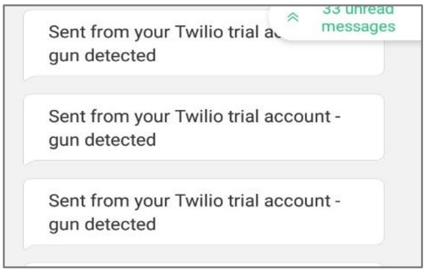
Fig 4.3: Alert message sent to concerned authorities

For the weapon detection module, we implement a forward pass for the current frame and collect bounding boxes from each grid. The final bounding box is chosen by the threshold minimum probability defined earlier by us (IOU threshold = 0.05). At the very end, if the targeted object, that is guns are detected in the frames, an alert is to be sent to a specified phone number of higher authorities, using third party software, Twilio.



(a) (b)

Fig 4.4 (a): Handgun is successfully detected; (b): AK-47 rifle is also successfully detected



Sent from your Twilio trial a gun detected

Sent from your Twilio trial account - gun detected

Sent from your Twilio trial account - gun detected

Fig 4.5: Alert message is sent to the specified number

Now for the integrated system output, both the modules are seamlessly working, and respective alerts are sent.

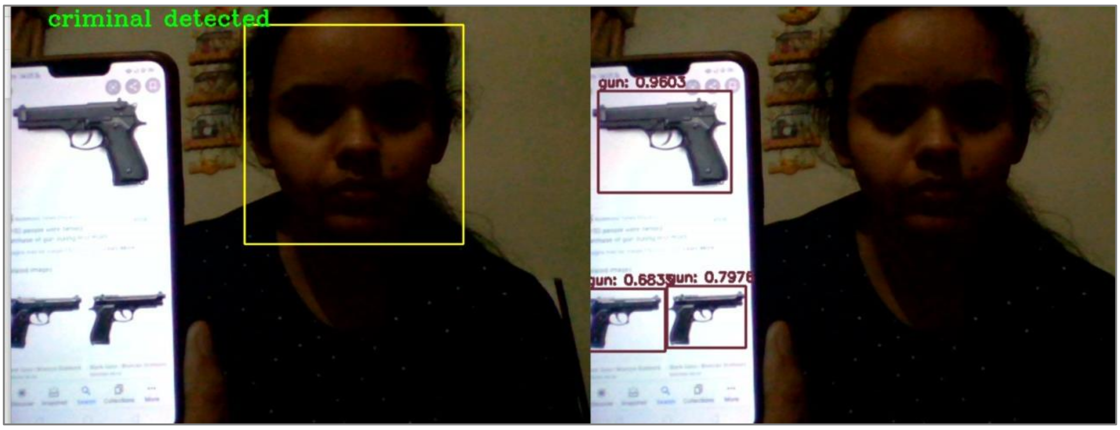
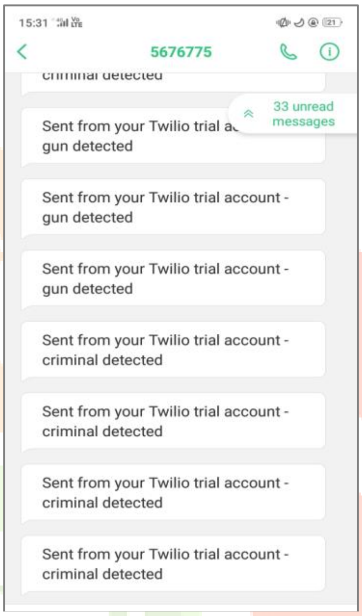


Fig 4.6: Concatenated output of the two models with respective predictions



criminal detected

Sent from your Twilio trial a… gun detected

Sent from your Twilio trial account - gun detected

Sent from your Twilio trial account - gun detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Sent from your Twilio trial account - criminal detected

Fig 4.7: Alert message sent for both criminal detection as well as gun detection

The accuracy of each of the individual models used for this proposed system, along with the dataset they have been trained on, has been specified in the table 4.1 below.

Table 4.1: Accuracy of individual machine learning models

Target	Machine learning model	Dataset	Accuracy
Face recognition	VGG16	100 images for each of the 8 people	97.66%
Gun detection	YOLOv4 with Darknet	7000 images of different types of guns	89.04%

V. Future Works

The proposed project can be further enhanced for the future, with the addition of more weapons to the dataset. The criminal dataset may be expanded for a real-life situation. Also, the sending of frames along with the alert will be more efficient and helpful for the higher authorities to take immediate action. Furthermore, features like human tracking can also be added to provide a more wholesome solution for surveillance system monitoring.

REFERENCES

[1] Hassan, Apoorva. P, Impana. H. 2019.Automated criminal identification by face recognition using open computer vision classifiers. Third International Conference on Computing Methodologies and Communication (ICCMC)

[2] L. Sirovich and M. Kirby. Low-Dimensional Procedure for the Characterization of Human Faces. Journal of the Optical Society of America, A 4 (1987): 519-524.

[3] M. Turk and A. Pentland.Eigenfaces for Recognition.Journal of Cognitive Neuroscience, Vol. 3, No. 1 (1991): 71-86 .

[4] Ajit U. Ushir, Vasanti R. Vishwakarma, Ajinkya A. Shete, Mahesh Sanghavi. 2013.Mathematical Modelling for Face Recognition System.International Conference on Recent Trends in Engineering & Technology.

[5] O. M. Parkhi, A. Vedaldi, A. Zisserman.2015.Deep face recognition. British Machine Vision Conference.

[6] Huapeng Yu, Zhenghua Luo, Yuanyan Tang.2016.Transfer Learning for Face Identification with Deep Face Model.7th International Conference on Cloud Computing and Big Data .

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E.2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems : 1097-1105.

[8] He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision : 2961-2969.

[9] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun.2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE transactions on pattern analysis and machine intelligence, 39(6): 1137–49.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 779– 788.

[11] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 7263– 7271.

[12] Joseph Redmon and Ali Farhadi. 2018.YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Apr 2020 . YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934v1 [cs.CV] 23.

[14] Raghunandan Srinath, Jayavrinda Vrindavanam, V. Prathith Vasudev, SupreethS, Harsh Raj, and Ananya Kesarwani.2021. A Machine Learning Approach for Localization of Suspicious Objects using Multiple Cameras. IEEE International Conference for Innovation in Technology, INOCON 2020, November 6th to 8th. IEEE Xplore: 01 January 2021 , DOI: 10.1109/INOCON50539.2020.9298364,Electronic ISBN:978-1-7281-9744-9 , Bengaluru, Karnataka, India.