Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

Fire and Gun Violence based Anomaly Detection System Using Deep Neural Networks

Parth Mehta
dept. Electrnonics and Telecomm.
Dwarkadas J. Sanghvi COE
Mumbai, India
parthmehta15@gmail.com

Atulya Kumar
dept. Electronics and Telecomm.
Dwarkadas J. Sanghvi COE
Mumbai, India
atulyakumar97@gmail.com

Shivani Bhattacharjee
dept. Electronics and Telecomm.
Dwarkadas J. Sanghvi COE
Mumbai, India
shivani.bhattacharjee@djsce.ac.in

***Abstract*— Real-time object detection to improve surveillance methods is one of the sought-after applications of Convolutional Neural Networks (CNNs). This research work has approached the detection of fire and handguns in areas monitored by cameras. Home fires, industrial explosions, and wildfires are a huge problem that cause adverse effects on the environment. Gun violence and mass shootings are also on the rise in certain parts of the world. Such incidents are time-sensitive and can cause a huge loss to life and property. Hence, the proposed work has built a deep learning model based on the YOLOv3 algorithm that processes a video frame-by-frame to detect such anomalies in real-time and generate an alert for the concerned authorities. The final model has a validation loss of 0.2864, with a detection rate of 45 frames per second and has been benchmarked on datasets like IMFDB, UGR, and FireNet with accuracies of 89.3%, 82.6% and 86.5% respectively. Experimental result satisfies the goal of the proposed model and also shows a fast detection rate that can be deployed indoor as well as outdoors.**

Keywords—YOLOv3, darknet, video processing, anomaly detection, deep learning, neural networks

I. INTRODUCTION

The main idea of our project is to create a system that monitors surveillance data of an area and sends alerts in case a fire or gun is detected. Closed Circuit Television (CCTV) cameras record video footage 24 hours of the day, however there isn't enough manpower to monitor each and every camera for various anomalous events. There are systems to detect fire using smoke sensors in many places like schools, educational institutes, etc. However, a cost-effective system that combines fire as well as gun detection for security purposes is the need of the time. Surveillance systems such as closed-circuit television (CCTV) and drones are becoming increasingly common. Research also shows that the installation of CCTV systems helps to combat mass shooting incidents [11] and are also extremely important for evidence collection.

The research work uses YOLO (You Only Look Once) object detection system [12] which uses convolution neural networks for object detection. It is one of the faster algorithms that performs without much degradation in accuracy.

The training of this model has been done on the cloud to save hundreds of hours of GPU time on a local runtime. Using hosted runtime has also been beneficial in fine-tuning our model to near perfection.

The guns and fires found in CCTV videos in the dataset occupy only a small portion of the entire frame, hence our primary objective is to implement an algorithm that would accurately draw multiple bounding boxes in such low-quality videos. Furthermore, the detection must be in real-time with relatively high accuracy as the scenario being processed could be time-sensitive. Also, there must low number of false positives since the authorities are being alerted once a detection above the threshold is made.

II. LITERATURE SURVEY

A paper proposing fire detection in video sequences was proposed by Celik et al. [1]. The system compares information about foreground objects with statistical color information of fire. A simple adaptive scene background model was developed by using three Gaussian distributions, each of which corresponds to the pixel statistics in the respective color channel. Using adaptive background subtraction algorithms, the foreground information is extracted and then verified by the statistical fire color model to decide whether or not the foreground object is a fire candidate.

The statistical fire color model consists of three rules. According to the first rule, the value of the red component of an RGB pixel must be greater than the mean of Red components of the entire image. The next rule states that the value of the red component of a pixel must be greater than the green component which must be greater than the blue component. The final rule takes into consideration the ratio of Red, Blue and Green components. All these rules complement the previous rules. Error is generated due to non-linearities in the fixed camera, sudden changes in lighting conditions and also due to some kind of materials producing different fire colors while burning. However, this method fails in case there is only smoke and no red-colored pixels.

Satellite-based systems can monitor a wide area, but satellite imagery resolution is low [2]. A fire is detected when it has grown quite a lot, so it is not possible to detect it in real-time. Such systems are also very expensive [3]. In satellite-based forest fire detection systems, weather conditions like overcast or rains strictly decrease the accuracy due to the limitations caused by the long scanning period and low resolution of satellites. [4]

M. Trinath et al. [5] propose an IOT based solution for the problem. Their system includes the use of temperature and smoke sensors. The biggest drawback of this system is that the sensors are costly and delicate and may be easily damaged due to various natural factors.

Celik et al. [6] proposed a novel model for detection of fire and smoke using image processing approach. Few rules are identified for fire pixels and then given to a Fuzzy Inference System (FIS) in the RGB and YCbCr color space. Based on the probability value, a rule table is formed depending on which a pixel is considered to be fire. They report to have 99% accuracy but, this cannot be used for real-time monitoring.

R-CNN based method for handheld gun detection [7] using a pre-trained VGG model is proposed. In the segmented images, Fast Retina Keypoint (FREAK) and Harris Interest Point Detector is used to find the weapons. Testing was done on a dataset built from the Internet Movie Firearm Database (IMFDB). The model could detect and classify three types of guns namely revolvers, rifles, and shotguns. However, for this method to detect guns, it has to be held by humans and not otherwise. The visual gun detection system using SIFT (Scale Invariant Feature Transform) and Harris interest point detector [8] was proposed which utilized color-based segmentation to take out a distinct object from an image using K-Means clustering algorithm.

Grega et al. [9] proposed a method for automatic detection of dangerous situations in CCTV systems, through the use of image processing and machine learning. Firearms and knives were detected in video using sliding window techniques, fuzzy classifiers and canny detectors. The dataset and detection system constructed by the authors were made available. [10].

III. Proposed Solution

The proposed experiment employs You Only Look Once (YOLO) v3 model, [13] which is a deep learning framework based on Darknet, an open-source neural network in C [14]. YOLOv3 is the best choice as it provides real-time detection without losing too much accuracy. The architecture used is darknet53 which consists of 53 convolutional layers each followed by Leaky ReLu activation and batch normalization layers, making it a fully convolutional network (FCN). For the task of detection, the total layers used are 106 which makes the model bulkier than its previous variants. The model doesn't use pooling to prevent loss of low-level features. Also, the unsampled layers are concatenated with the previous layers to help detect small objects by preserving the minute features. Unlike the sliding window and region proposal-based techniques, YOLO detects objects in an image very well as it gets every detail about the whole image and the object by seeing the entire image.

The image is divided into grids and N bounding boxes with confidence scores are predicted using image classification and localization on each grid cell. YOLO does detections on three different scales ranging from small to large at layer number 82, 94 and 106. The larger objects are detected by 13 x 13 layer, medium objects by 26 x 26 layer and smaller objects by 52 x 52 layer.

The model predicts 4 coordinates for each bounding box, t_x, t_y, t_w, t_h. The cell is offset from top left corner of the image by coordinates (c_x, c_y). The prior bounding box has a width and height p_w, p_h. The predictions b_x, b_y, b_y *and* b_h correspond to:

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

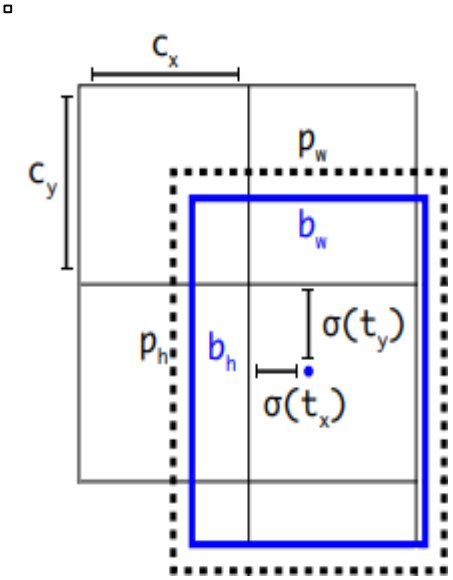

Fig. 2. Bounding boxes with location prediction and dimension priors. The height and width of the box is predicted as offsets from cluster centroids. The center coordinates of the box are predicted relative to the location of filter application using a sigmoid function.

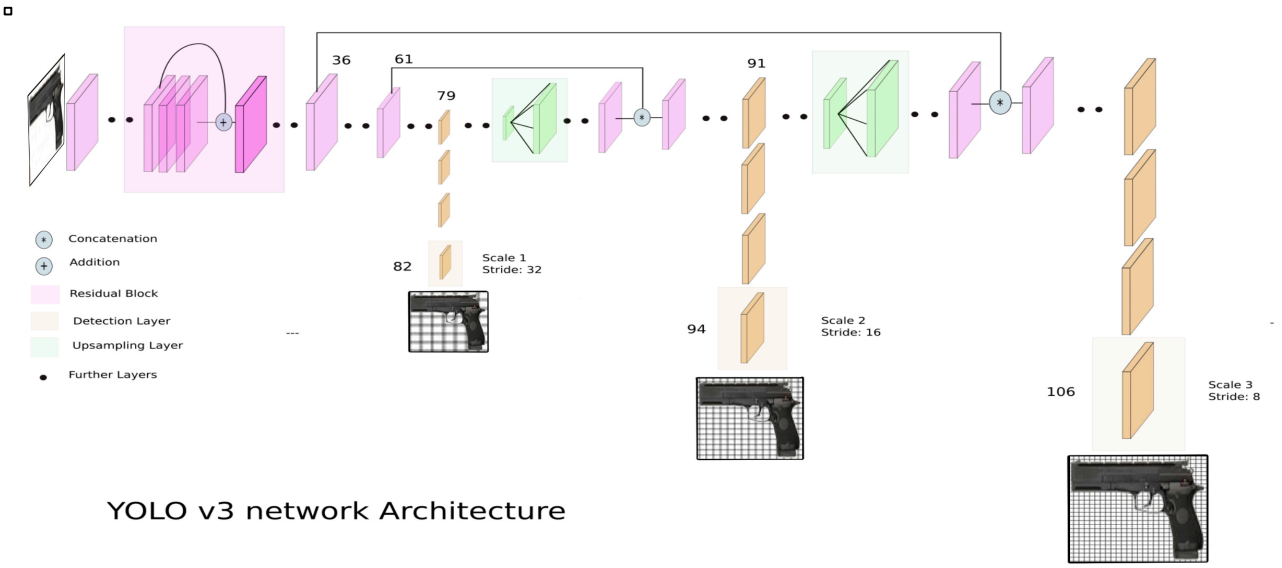

Fig. 1. YOLOv3 Network Architecture.

The loss function of YOLOv3 is:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \begin{bmatrix} (t_x - \widehat{t_x})^2 + (t_y - \widehat{t_y})^2 \\ +(t_w - \widehat{t_w})^2 + (t_h - \widehat{t_h})^2 \end{bmatrix}$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \left[-\log(\sigma(t_o)) + \sum_{k=1}^{C} BCE(\widehat{y_k}, \sigma(s_k)) \right]$$

$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{noobj} \left[-\log(1 - \sigma(t_o)) \right] \quad (1)$$

Assuming Prediction vector: $t_x, t_y, t_w, t_h, t_o, s_1 \dots s_c$ and corresponding ground truth label: $\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h, \hat{t}_o$, $\hat{y}_0, \hat{y}_1, \dots \hat{y}_c$ where c = total classes. $y \in \{0,1\}$.

Lambda constants are used to differentially weight the loss function to improve model stability. Highest penalty is for coordinate prediction with $\lambda_{coord} = 5$ and lowest penalty for when no object is present i.e. $\lambda_{noobj} = 0.5$

'σ' is the sigmoid function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Binary Cross Entropy is defined as:

$$BCE(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} y.\log(\hat{y}_i) + (1 - y).\log(1 - \hat{y}_i)$$

y is the actual value and $\hat{y}$ is the predicted value. BCE measures the error in the actual and predicted values.

To ensure focus training on boxes containing an object, we set the following mask:

$$1_{ij}^{obj} = \begin{cases} 1 & \text{if the object exists in the i} - \text{th cell and} \\ & \text{j} - \text{th box is responsible for detecting it} \\ 0 & otherwise \end{cases}$$

$$1_{ij}^{noobj} = \begin{cases} 1 & \text{if there is no object in i} - \text{th cell} \\ 0 & otherwise \end{cases}$$

Here, the squared error terms have been replaced by cross-entropy error terms i.e. class predictions and object confidence are now being predicted through logistic regression.

Non-Maximal Suppression (NMS) is used to detect the best bounding box out by first rejecting predicted bounding boxes that have a detection probability that is less than a given NMS threshold and then eliminating all the bounding boxes whose Intersection Over Union (IOU) value is higher than a given IOU threshold. Intersection over Union is an evaluation metric used to measure the accuracy of an object detector.

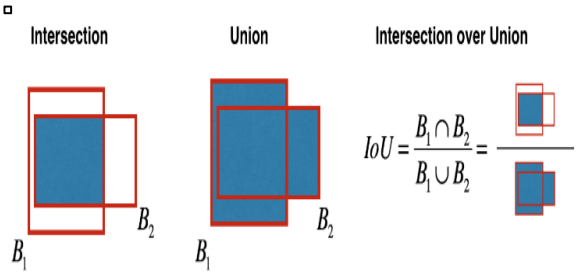

Fig. 3. Intersection over Union is computed simply by dividing the area of overlap by the area of their union of the bounding boxes.

This step will remove all boxes that have a large overlap with the selected boxes. Only the "best" boxes remain.

The performance of YOLO v3 is at par with other state-of-the-art detectors like RetinaNet while being considerably faster, at Common Objects in Context (COCO) with mAP 50 benchmarks. It is also better than Single Shot Detector (SSD) and its variants. It gives 30-45 frames per second output on a real-time video on a GPU.

IV. Dataset Description

Our model was trained using 3000 images of guns from the UGR handgun dataset [15]. The dataset contains images of guns in a variety of different angles, positions and orientations. The dataset is also annotated in *(<x_{min}> <y_{min}> <x_{max}> <y_{max}>)* format which were converted into the format suitable for YOLO *(<object-class> <x_center> <y_center> <width> <height>)*.

500 images containing fire were also used which were downloaded from Google and were annotated by using *LabelImg* - a graphical image annotation tool [16].

For testing gun detection, we are using the UGR handgun testing dataset and the IMFDB dataset [17]. The IMFDB dataset contains around 4000 images of various guns, rifles, shotguns, etc. These are images of various movie scenes. The negative images in the dataset contain images of objects with shapes similar to a gun like hairdryers, drills, etc. For testing our model's performance on images containing fire we are using the images in the FireNet Dataset [18]. We have also made a custom dataset of 19 images downloaded from Google which contains images of people holding handguns from a CCTV perspective and various other angles as well as adding few videos from Gun movies database [10] and FireNet Dataset. Our dataset is called Fire-Gun dataset [19].

V. Implementation

Our model has been trained for the parameters of YOLO as shown in Table I. Guns are represented by label 0 and Fire by label 1. Fig. 4 represents the training graph.

Table I. Training Parameters

Parameters	Description
Max Batches	4000
Steps	3200, 3600
Filters	21
Classes	2

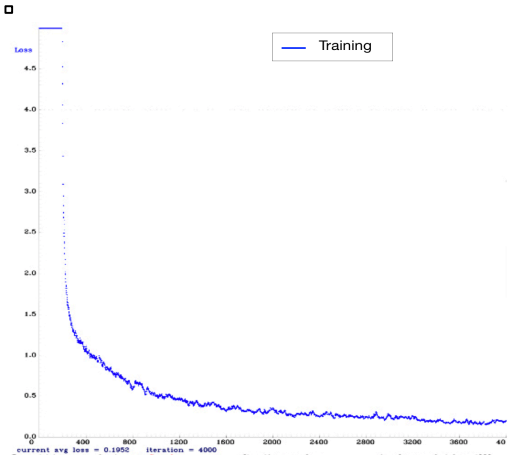

Fig. 4. Training graph for 4000 iterations

After training for 4000 iterations the loss is 0.2864. The loss is calculated using Eq. 1.

VI. Output

Each prediction takes around 0.7 seconds when running on CPU and 44 milli-seconds on GPU.

A. Performance on UGR Dataset

The UGR handgun testing dataset contains 304 images with guns and 304 images without a gun. The performance of the model excellent on the UGR testing dataset as can be seen from Table II. This is also due to the fact that the positive images in the dataset contain focused images of handguns only.

Table II. Confusion Matrix of Model on UGR Dataset

	Actual Positive	**Actual Negative**
Predicted Positive	303	64
Predicted Negative	1	240

Fig. 5. Prediction on an image in UGR dataset.

B. Performance on IMFDB Dataset

The IMFDB dataset contains around 4000 positive images with a gun and around 2000 negative images without a gun. However, we will be testing it only on around 400 images of each class.

Table III. Confusion Matrix of Model on IMFDB Dataset

	Actual Positive	**Actual Negative**
Predicted Positive	390	92
Predicted Negative	51	289

The performance of the model is not as good as on the UGR dataset as seen from the confusion matrix in Table III. This is because there are a lot of images with rifles and shotguns for which our model has not been trained. The proposed model is only trained for handguns. Also, the negative images are of objects very similar to the shape of a gun like hairdryers, etc. which can be seen in Fig. 6.

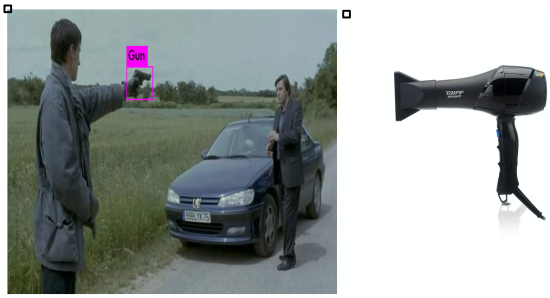

Fig. 6. The image on left shows our model prediction on a positive image and the image on right is a negative example on an image in IMFDB dataset.

C. Performance on FireNet Dataset

From the FireNet dataset, we select 200 images with and without fire for testing the fire detection capability of our model. The performance of the model can be seen in Table IV. Fig. 7 shows a few examples of fire detection by the proposed model.

Table IV. Confusion Matrix of Model on FireNet Dataset

	Actual Positive	**Actual Negative**
Predicted Positive	159	13
Predicted Negative	41	187

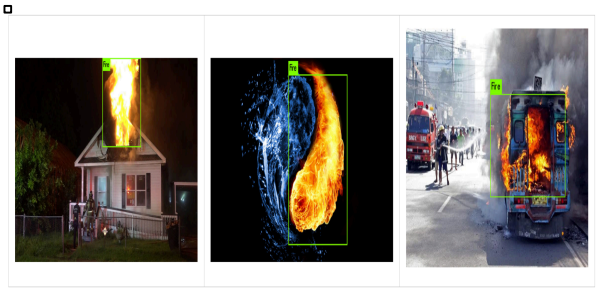

Fig. 7. Prediction on an image in FireNet dataset.

D. Comparison of model performance on different datasets

In this section, we compare the performance of our model on the UGR, IMFDB, and FireNet datasets. The comparison can be seen in Table V. The values of Accuracy, Precision, Recall, and F1 Score are calculated from values in Table II, Table III and Table IV using Eq. 2, Eq. 3, Eq. 4, and Eq. 5 respectively.

$$Accuracy = (TP + TN) / (P + N) \quad (2)$$

$$Precision = TP / (TP + FP) \quad (3)$$

$$Recall = TP / (TP + FN) \quad (4)$$

$$F1\ Score = 2*(Precision* Recall)/(Precision+Recall) \quad (5)$$

Table V. Comparison of Model Performance on Different Datasets

	UGR	IMFDB	FireNet
Accuracy	0.8931	0.8260	0.8650
Precision	0.8256	0.8091	0.9244
Recall	0.9967	0.8844	0.7950
F1 Score	0.9031	0.8451	0.8548

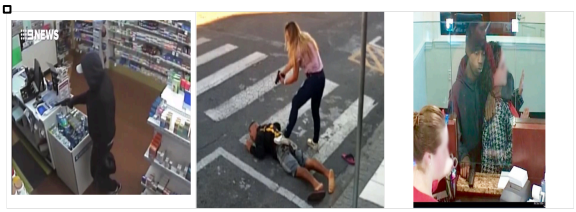

Fig. 9. Some images where the model fails to identify gun.

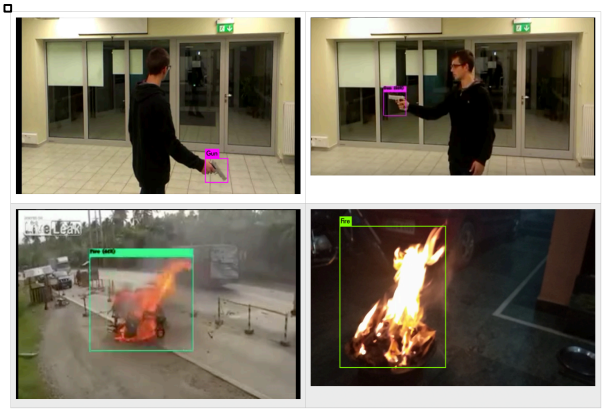

Fig. 10. Video frames with fire or gun predictions.

E. Model Performance on custom Fire-Gun Dataset

A custom dataset has been created as there was no dataset for images of guns from a CCTV perspective. Therefore, 19 gun images were collected with various angles, many of which are CCTV images of humans with a gun. Our dataset also includes 4 videos containing gun and fire to test the performance of our model on video. Fig. 8 shows the successful predictions of our model on images in our dataset and Fig. 9 shows images where our model fails to detect guns.

The model also gives good performance when used on video by frame by frame processing. Fig. 10 shows the frames of video in our dataset where there are fire or gun predictions.

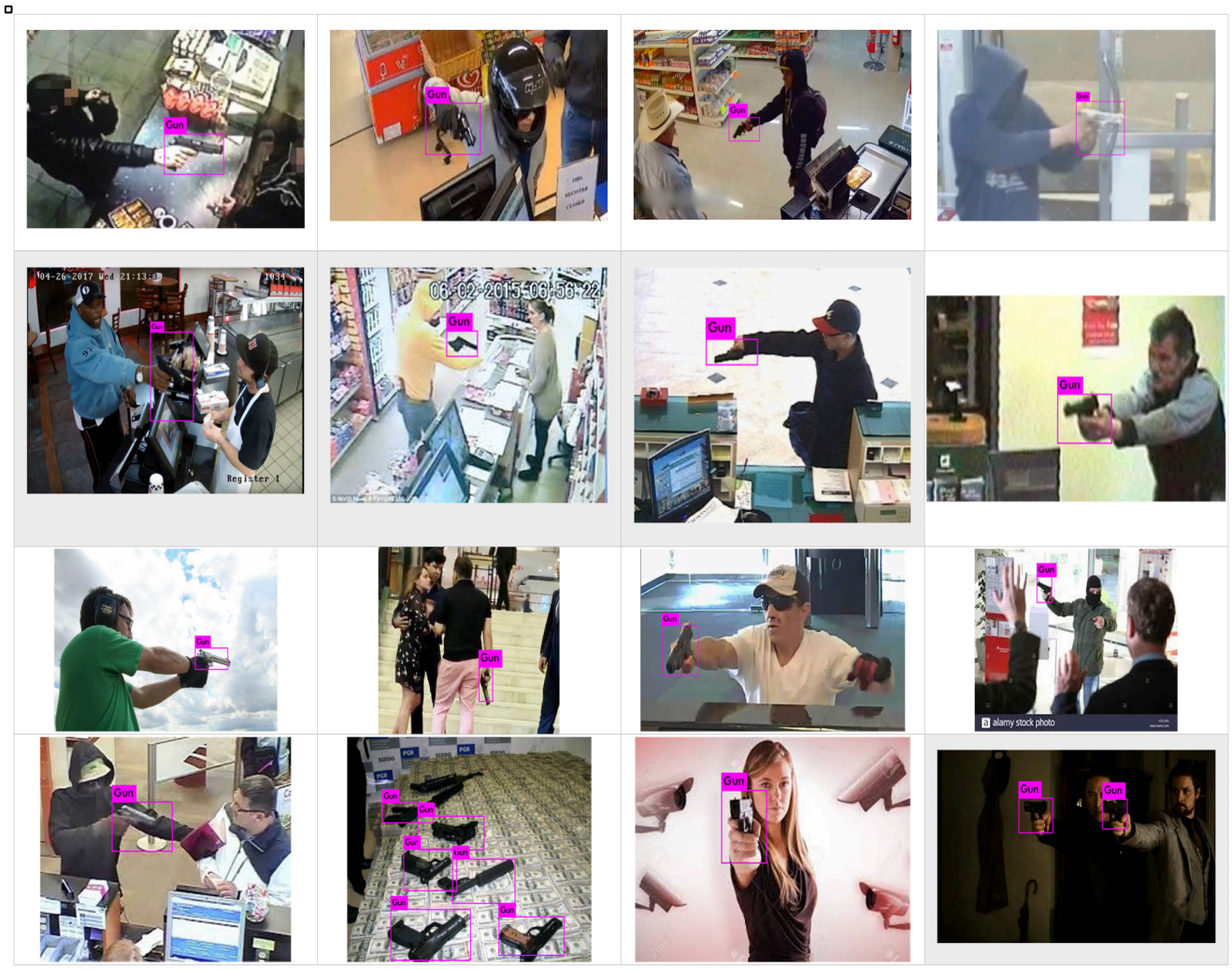

Fig. 8. The performance of the model on images of our custom dataset. It can be seen that the guns are in a variety of different orientations and positions.

VII. Applications

1) Instant large-scale surveillance footage processing: Hundreds of gigabytes of surveillance footage are recorded in a city each day however having large amounts of unprocessed video data is inconsequential and would require a lot of people to manually process these.

2) Alerting the authorities in case of an anomaly: The proposed system can immediately detect guns or fires in public places as well as inside buildings, and alert the fire brigade or police or both. The model can also be deployed in fire prone areas like camp-sites, forests, oil refineries and factories with explosive chemicals as well as in educational institutions.

3) Evidence Collection: The system also proves useful for evidence collection by forensics and also determine the exact location of the perpetrator based on the camera location. This helps to greatly speed up the judicial process.

VIII. Conclusion

In this paper, a real-time frame-based efficient fire and gun detection deep learning model has been presented with a high accuracy metric. The Darknet53 model might be bulky but has a good detection capability. The detections per frame are appropriate for real-time monitoring and can be deployed on any GPU based system.

IX. Future Scope

The gun dataset can be expanded by adding annotated images of shotguns and rifles to make the model more robust. There is also scope for the model to recognize different types of guns based on the small variations. The above model only detects fire. In the future, this system can be combined along with a fire suppression system. The system can include water sprinklers or fire extinguishers.

References

[1] T. Celik, H. Demirel, H. Ozkaramanli and M. Uyguroglu, "Fire Detection in Video Sequences Using Statistical Color Model," 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, 2006, pp. II-II.

[2] B. U. Toreyin, Y. Dedeoglu and A. E. Cetin, "Flame detection in video using hidden Markov models," IEEE International Conference on Image Processing 2005, Genova, 2005, pp. II-1230.

[3] Z. Li, S. Nadon and J. Cihlar "Satellite-based detection of Canadian boreal forest fires: Development and application of the algorithm," International Journal of Remote Sensing, vol. 21, no.16, pp. 3057-3069, 2000.

[4] T. J. Lynham, C. W. Dull and A. Singh, "Requirements for space-based observations in fire management: a report by the Wildland Fire Hazard Team, Committee on Earth Observation Satellites (CEOS) Disaster Management Support Group (DMSG)," IEEE International Geoscience and Remote Sensing Symposium, Toronto, Ontario, Canada, 2002, pp. 762-764 vol.2.

[5] M. T. Basu, R. Karthik, J. Mahitha, and V. L. Reddy, "IoT based forest fire detection system," International Journal of Engineering & Technology, vol. 7, no. 2.7, p. 124, 2018.

[6] T.Celik, H.Demirel, H.Ozkaramanli, "Fire and Smoke Detection without Sensors: Image Processing Based Approach," Proceedings of 15th European Signal Processing Conference, Poland, September 3-7, 2007.

[7] G. K. Verma and A. Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning," Proceedings of the 7th International Conference on Computer and Communication Technology - ICCCT-2017, 2017.

[8] R. K. Tiwari and G. K. Verma, "A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector," Procedia Computer Science, vol. 54, pp. 703–712, 2015.

[9] M. Grega, S. Łach and R. Sieradzki, "Automated recognition of firearms in surveillance video," 2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), San Diego, CA, 2013, pp. 45-50.

[10] "Guns movies database," Katedra Telekomunikacji *AGH*. [Online]. Available: http://kt.agh.edu.pl/grega/guns/. [Accessed: 30-Mar-2020].

[11] R. Kayastha, "Preventing Mass Shooting Through Cooperation of Mental Health Services, Campus Security, and Institutional Technology," 2016.

[12] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788.

[13] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," arXiv 2018 arXiv:1804.02767.

[14] J. Redmon , "Darknet: Open source neural networks in c," 2013.

[15] "UGR Handgun Dataset," Weapons Detection | Soft Computing and Intelligent Information Systems. [Online]. Available: https://sci2s.ugr.es/weapons-detection. [Accessed: 19-Mar-2020].

[16] "LabelImg Annotation Tool," GitHub, 30-Jan-2020. [Online]. Available: https://github.com/tzutalin/labelImg. [Accessed: 30-Mar-2020].

[17] R. Kanehisa and A. Neto, "Firearm Detection using Convolutional Neural Networks," Proceedings of the 11th International Conference on Agents and Artificial Intelligence, vol.2, pp. 707–714, 2019.

[18] "FireNet Dataset," *GitHub*, 11-Dec-2019. [Online]. Available: https://github.com/arpit-jadon/FireNet-LightWeight-Network-for-Fire-Detection. [Accessed: 19-Mar-2020].

[19] "Fire-Gun Dataset," *Kaggle*, 18-Mar-2020. [Online]. Available: https://www.kaggle.com/parthmehta15/fire-gun. [Accessed: 30-Mar-2020].