2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)
Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. Aug 18-19, 2023

Weapon Detection from Surveillance Footage in Real-Time using Deep Learning

Vedantika Jadhav
Computer Department
Pimpri Chinchwad College of Engineering
Pune, India
vedantika.vj26@gmail.com

Rutuja Deshmukh
Computer Department
Pimpri Chinchwad College of Engineering
Pune, India
rutujad252@gmail.com

Palak Gupta
Computer Department
Pimpri Chinchwad College of Engineering
Pune, India
palakgupta6506@gmail.com

Sharvi Ghogale
Computer Department
Pimpri Chinchwad College of Engineering
Pune, India
gnsharvi@gmail.com

Mahalakshmi Bodireddy
Computer Department
Pimpri Chinchwad College of Engineering
Pune, India
mahalakshmi.bodireddy@pccoepune.org

2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA) | 979-8-3503-0426-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCUBEA58933.2023.10392003

Abstract— **Weapon violence is a serious threat nowadays due to the rise in technology and criminal intelligence. Manual surveillance is a very tedious task because these activities are atypical in comparison with everyday activities. By introducing machine learning techniques to detect such activities, we can minimise the risk of human errors and prevent details from going unnoticed. Current systems fail when it comes to efficiency of the system and ease of use. We have implemented various models like CNN, YOLOV7, YOLOV8 and VGG for weapon detection and identified their strengths and limitations. A complete system for detection of weapons using these models and raising an alarm has been implemented here to overcome the problems of manual surveillance. Results are presented in a table focusing on metrics like precision and recall as these metrics prove to be more insightful and reliable as compared to accuracy for object detection.**

Keywords—object detection, deep learning, gun detection, YOLO, AATpI, CNN

I. Introduction

Suspicious activities are a problem in urban as well as suburban areas with the increase in population, technology and criminal intelligence. Surveillance cameras are a great way to monitor vast, crowded and critical areas at once.

However, manually surveilling the data may lead to human error and crucial information could go unnoticed, which is a risky job. Thus, with the help of machine intelligence, surveillance tasks to detect suspicious activities such as weapon detection, abandoned luggage detection etc can minimise the risk of human error and even save lives. We focus on analysing cases, with the help of deep learning technologies, those if ignored may be a cause to criminal activity and even include a high risk of human lives. We also discuss technology, which will help raise an alarm or alert when such activity is discovered.

Many surveillance cameras are used to trace a criminal activity but not to detect and minimise such activities from happening. In today's world, which is growing fast with new technologies, methods and criminal intent also witnessing a dangerous rise, it is important to keep a constant watch in areas which may be prone to such activities. As the crime rates increase, it becomes necessary to identify them in time and take necessary precautionary actions. In almost all cases, this task resides with a human or a group of humans. This increases the chances of human errors and crucial details going unnoticed. Surveillance cameras are a great way to monitor vast, crowded and critical areas at once. However, manually surveilling the data may lead to human error and crucial information could go unnoticed, which is a risky job. Thus, with the help of machine intelligence, surveillance tasks to detect suspicious activities such as weapon detection, abandoned luggage detection etc can minimise the risk of human error and even save lives. However, with machine intelligence, human errors can be avoided and details will not be missed.

The main objective of this paper is to provide a thorough analysis of different object detection models through various metrics like precision, recall and F1 score. The survey consists of the four most popular deep learning object detection models namely Faster RCNN, RFCNN, SSD and YOLO. This analysis will further assist the user to determine the best model that fits their technical requirements. After learning these models, we have implemented four models, which are CNN, VGG, YOLOV7 and YOLOV8 and compared their results. We have used a single class dataset that consists of 2971 images of guns. Using the model that provides the best performance, we have created a web application that can detect guns from real time videos and footage.

II. Literature Review

Many activities observed under surveillance footage may pose a certain level of threat to the lives of the people. Our research was based mostly on weapon detection as a part of suspicious activity. Many times it is observed that weapons are taken out first before anything can be done with it. This important detail may go unnoticed during manual surveillance. Our focus on detecting weapons is on guns. When it comes to detecting a weapon in an image or a video, object detection and object recognition come into picture. The best techniques for detecting objects in an image or a video are SSD and RCNN. Machine learning models such as Region Convolutional Neural Network (RCNN)[1][2] and Single Shot Detection (SSD)[1] are considered best approaches for object detection.

For detecting weapons, search techniques are required as a first step. Selective search techniques are applied to generate candidate regions where there is a high possibility of the object to be located. There is one classical search technique called a sliding window. Around millions of candidate windows are generated which makes it very slow. Hence this approach is not suitable for applications where we require to detect objects in real time footage. There is one more technique called Region Proposal (RP) search algorithm that is the fastest search technique[3]. It generates much less candidate regions. The first model which introduced the RP search algorithm was RCNN, which was later made into a fully convolutional network. Region Proposal Network (RPN) is a modified version of RP search technique[3]. It allowed conversion of two step detection models into one step. This approach was first adopted by Faster RCNN followed by RFCN[3].

Convolutional neural networks are constructed as a stack of learnable layers (convolutional and fully-connected layers) and reduction layers (pooling layers). A widely-used neural network architecture known as Residual Networks (ResNet) serves as the backbone for numerous computer vision applications. ResNet50 is composed of 50 and ResNet101 is of 101 learnable layers[3].

To detect objects such as guns in an image in the dataset [2], the RCNN approach works for the generation of region proposals and a network for the detection of objects. This method makes use of two RPN networks to predict object boundaries at specific positions. Region proposals and object detection are separately done with the two RPN networks[3]. For object boundaries, anchors are defined with a selective search algorithm. TensorFlow implementation of Faster RCNN [1] is used for basic detection of objects from the MS-COCO dataset.

Another method, which proved to be more effective [2] is the SSD approach. This approach discards the idea of having regional proposals, which gives better results in case of speed even in low resolution images with the use of multi-scale features[2][4]. However, fast RCNN gives better accuracy. RFCN is an improved version of Faster RCNN. It combines the RPN and classifier model to re-utilize calculation and memory access[2][3].

YOLO (You Only Look Once) is a popular object detection system that has been used for various applications, including weapon detection. YOLOv7 is an implementation of YOLO that has been optimised for both accuracy and speed, making it a good choice for real-time weapon detection. To detect weapons using YOLOv7, the first step is to train the network on a dataset of images that contain different types of weapons. The network learns to recognize the patterns and features that are unique to weapons and can then use this knowledge to detect them in new images. Once the network is trained, it can be used to detect weapons in real-time by processing video streams or images. The process involves dividing the input image into a grid of cells, with each cell assigned to detect any objects that intersect with it. YOLOv7 uses a single neural network for predicting class probabilities, bounding boxes and confidence scores for each object detected in the image. To detect weapons specifically, YOLOv7 can be trained on a dataset that includes images of various types of weapons, such as handguns, rifles, and knives. The network learns to recognize the unique features of each weapon type and can then use this knowledge to detect them in real-time. To improve the accuracy of weapon detection, YOLOv7 can be fine-tuned on specific datasets that contain images of weapons in different environments, such as indoor and outdoor environments. Fine-tuning the network on such datasets can improve its ability to detect weapons in challenging conditions, such as low light or cluttered scenes.

There exist various techniques for object detection that form the basis of artificial intelligence, with one of the most prominent being the You Only Look Once (YOLO)[8][7] algorithm and an up to date version of it. It shows us the comparative study of YOLO, YOLO V2, YOLO V3, YOLOV4, and YOLO V5. YOLO and YOLO V2, are not effective when it comes to the detection of small targets or detection on a multi-scale. YOLO V3[9] has multi-scaling features for object detection and adjusts the structure of the primary network, which overcomes the limitations of the previous algorithm[10]. Then comes the YOLO V4, which has a huge difference and focuses on comparing data showing considerable improvement. YOLO V5 is more convenient to use because of its multiple network architecture, has a very lightweight model size and its accuracy is as good as YOLO V4.

YOLOV7 and V8 are the fastest object detection algorithms available for new era machine learning problems. These algorithms surpass all previous YOLO models in terms of speed and accuracy. Training time required for these models is also less as compared to their previous counterparts. While comparing V7 and V8, it is often concluded that V8 outperforms all its previous versions, however the training time for V8 is an issue.

An alarm can be enabled in the detection system if there is certainty regarding the suspicious issues inside the scene. Alarm Activation time per Interval (AATpI) metric is used in alarm systems[3]. This approach counts the time it takes the system to locate k-successive images of true positives to trigger an alarm.

In general, steel weapons have reflective surfaces which under various conditions of brightness, deform and blurs their shapes within the frame. Darkening and Contrast at Learning and Test time (DaCoLT)[5] is a pre- processing method used to reduce the effects of weapon reflections. It uses data augmentation to test the detection model with respect to a specific scale of brightness conditions. It applies the darkening factor on frames to adjust contrast and to detect objects more accurately.

The development of anomalous activity detection systems aims to enhance the intelligence of surveillance systems. The goal is to notice suspicious or anomalous incidents in footage to avoid any accident or to issue alerts when a suspicious problem occurs. This problem mainly focuses on automatically detecting anomalous activities without much human intervention[5]. It describes a technique which automatically detects various suspicious activities from videos captured by cameras. It works in 3 steps : object detection, feature extraction and dominant behaviour analysis. Features such as centroid, speed, dimensions and directions are extracted for analysis. These features are useful for tracking objects within video frames. The only limitation is that they have considered 3 activities only and there are much less people moving in the video. But in surveillance cameras many people are moving from one place to another. Hence we need to track the activity of every person.

III. Methodology

A. System Architecture

Based on our research, we have discovered that the Weapon Detection system is currently limited to pre-recorded datasets and is not actively used in security cameras. To implement the system, the model must be trained, tested, and given access to a camera. With the rising crime rates, timely identification of potential threats is crucial. Figure 1 describes the network architecture used for learning and detecting guns in an image with the help of annotations.

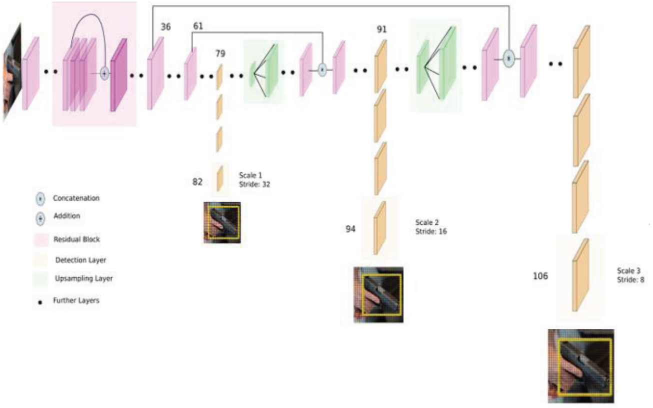


Fig. 1. . System Architecture

B. Deep Learning based Models

1) YOLO

The approaches to object detection in YOLO v7 and v8 differ significantly. YOLO v7 utilises anchor boxes and a new loss function known as "focal loss," enabling it to identify a broader range of object shapes and sizes than its predecessors. This, in turn, helps minimise the number of false positives. In contrast, YOLO v8 delivers the best performance among all three versions. For a comparable runtime, all v8 models exhibit a mAP improvement of +4 to +9 over v5. The new API in YOLO v8 simplifies both CPU and GPU device training and inference and supports previous YOLO versions.

YOLO v7 and v8 repositories include a detection file test file, which are necessary for detection of objects in the model and testing the performance of the model, respectively. For training, it makes use of a .pt file named ‘yolo_training.pt’. The dataset was divided into training, validation and testing, mentioned further in detail below. With a batch size of 16 and 55 epochs, yolov7 and v8 models were trained.

The file test.py produces the required performance metrics and confusion matrix after the model is trained. Based on these performance metrics, which are precision, recall, mean average precision and f1 score, the models were evaluated and the best one was chosen.

2) CNN

To implement CNNs for weapon detection, the first step is to collect and prepare a dataset of images containing different types of weapons. This dataset is then divided into training, validation, and testing sets. Next, the CNN is designed by defining the architecture of the network, including the number and types of layers, the activation functions used, and the number of nodes in each layer.

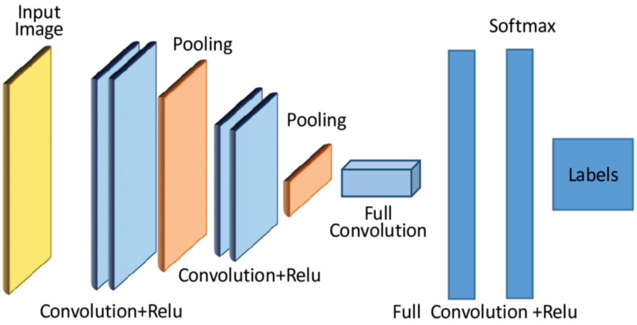


Fig. 2. CNN Architecture

For our research, we used a total of 5 convolutional layers in sequential format. For the first 3 layers, the activation function used was The rectified linear unit (ReLU) activation function, which adds non-linearity to a deep learning model and addresses the problem of vanishing gradients. For the last layer, a sigmoid activation was used which allows the network to introduce non-linearity into the model. Finally the CNN model was trained with 18 epochs and a batch size of 342.

During training, the network learns to identify the unique features of each type of weapon and use this knowledge to accurately detect weapons in new images. After training, the performance of the network was evaluated on the validation dataset to assess its accuracy. Finally, the trained CNN was applied to new images or videos to detect weapons in real-time. The input image is processed by the CNN, which generates a heat map of potential weapon locations. This heat map is then thresholded to generate bounding boxes around the detected weapons.

3) VGG

VGG-16 is a widely recognized convolutional neural network (CNN) architecture that is popularly utilised in computer vision tasks like object classification and detection, including weapon detection.

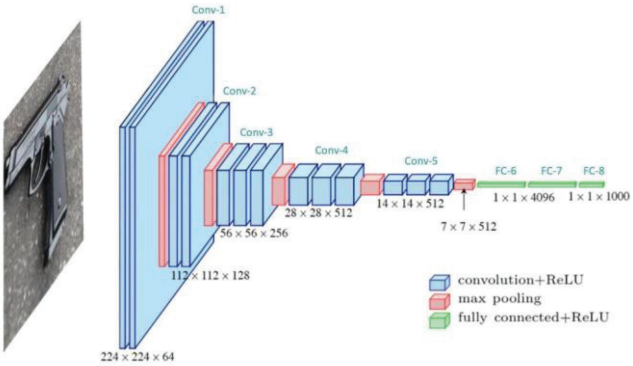


Fig. 3. VGG-16 Architecture

We modified the last fully connected layer of the VGG-16 network to output two classes: one for weapons and another for non-weapons. To avoid overfitting, we froze the weights of all layers except for the last few fully connected layers. We then trained the modified VGG-16 network on the training set using a loss function of binary cross-entropy and an optimizer of stochastic gradient descent. Once the network is trained, we evaluate its performance on the validation set to tune the hyperparameters and prevent overfitting. Finally, we applied the trained VGG-16 network to new images to detect weapons in real-time. The input image is processed by the VGG-16 network, which generates a heat map of potential weapon locations. This heat map is then thresholded to generate bounding boxes around the detected weapons. To improve the performance of the VGG-16 network for weapon detection, we can use transfer learning techniques such as fine-tuning. In fine-tuning, the pre-trained VGG-16 network is further trained on a smaller dataset of images containing weapons to learn to detect the unique features of each weapon type and improve its accuracy in detecting them.

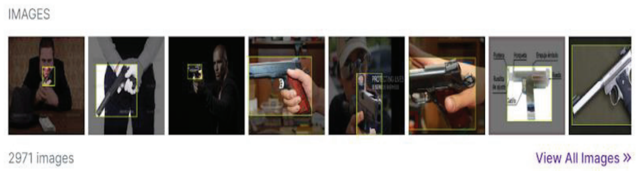
C. *Experimental Setup*

In this system, we have emphasised on detection of weapons in real time and sending an alert to the respective authorities. The weapon detection application consists of a GUI that enables the user to detect weapons from a realtime CCTV camera. In addition to this, the user can also capture snapshots or record the incoming video stream and store it in a database for future reference.

During the training of our models, we utilised the NVIDIA TESLA P100 GPU, which is designed specifically for deep learning applications and gives outstanding performance. Its advanced features are essential for optimising the training of complex neural networks like YOLO.

1) *Dataset preparation*

In order to create a database that enables the detection model to effectively differentiate guns from objects that may bear a resemblance, we begin with data preparation using LabelImg tool. For yolov8, the format must be txt.



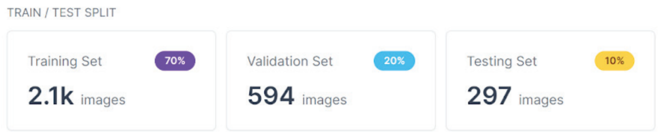
IMAGES

2971 images

View All Images »

Fig. 4. Dataset Preparation

For efficient testing, the images have been split into train.txt and test.txt. The .txt files contain the path of train and test images in our data set line by line. The dataset that we used contains approximately 2.2k images and it is further split into train, test and validation sets.



TRAIN / TEST SPLIT

Training Set 70%	Validation Set 20%	Testing Set 10%
2.1k images	594 images	297 images

Fig. 5. Dataset Split

For our experiments, we have used Keras API 2.0.4. The performance evaluation metrics, namely precision, recall, and F1 score, were computed for the classification model that was trained on datasets. The corresponding values for these metrics are presented in Table 1. Where,

Precision = TP/ (TP + FP)

Recall = TP/ (TP + FN)

F1 score = 2× (Precision × Recall) / (Precision + Recall)

Fig. 6. Example images from dataset

2) *Input*

The input to our application can be any video stream that has been captured from either a CCTV surveillance camera or an external webcam. There are no limitations on the format of the input video.

3) *Database*

The application has various features to enable detection of weapons and storage of the incoming input video in different formats like .jpg for storing snapshots from the video, and .avi format for recording particular parts of the video as and when required. These pictures and videos are stored in a database and can be used for future analysis.

4) *Filters*

The application also consists of multiple filters that can be applied for efficient processing and viewing of the input video in case of poor light conditions or limited resources. The video can be converted to greyscale or negative.

5) *Alarm System*

An alarm system is embedded that notifies the authorities in case a weapon is detected. A confidence score is provided that acts as a threshold value. The model sounds the alarm when an object is detected whose score is above this threshold value.

We constructed and evaluated the detection models by utilising several programming tools and libraries, including Tensorflow, PyCharm, Python CV2 APIs. Specifically, we used the Tensorflow framework for the development of the models, PyCharm for the code editing and debugging, and Python CV2 APIs for the image and video processing tasks.

D. *Selecting the most effective detection model*

We implemented 4 different models in order to identify the most effective model for weapon detection in real time. Implemented models:

- YOLOv7
- YOLOv8

- CNN
- VGG

The above models were trained and tested on a dataset that consisted of approximately 3000 images. The efficiency of these models was measured based on various performance metrics such as accuracy, true positives, false positives, precision, recall, and F1 score. Measures of performance metrics are given in Table I.

We base our conclusions on the F1 score of all the models. A higher precision and recall result in a higher F1-score. The F1-score is normalised and falls within the range of 0 to 1, with higher scores indicating better model performance.

IV. Results and Discussion

TABLE I. COMPARISON OF MODELS ON THE BASIS OF PERFORMANCE METRICS IN PERCENTAGE

	YOLOv7	YOLOv8	CNN	VGG
PrecisioN	89.9	87.4	79.24	88.26
RecalL	75.9	81.8	60.00	63.42
F1-score	82	89.1	68.78	77.34
mAP	86.1	87.7	60.34	84.70

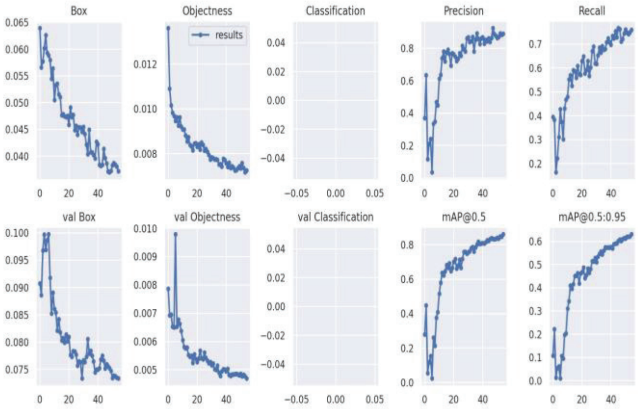


Fig. 7. YOLOv8 Results

Based on the results obtained, it can be concluded that YOLOv7 and YOLOv8 are both effective in detecting objects in images with good precision and recall rates. However, YOLOv8 outperformed YOLOv7 with a slightly higher mAP and F1 score. The CNN model achieved high accuracy but relatively lower precision and recall rates compared to YOLOv7 and YOLOv8. On the other hand, the VGG model had the highest precision rate and a good recall rate but with a lower accuracy rate compared to CNN. Overall, the YOLOv8 model is recommended as the best detection model for the dataset used in this study.

When evaluating object detection algorithms, precision and recall carry more significance than accuracy. As such, these metrics were used to test the performance of various algorithms. Of all the tested algorithms, Yolov8 demonstrated superior performance with an F1-score of 89.1% and a mean average precision of 87.7%.

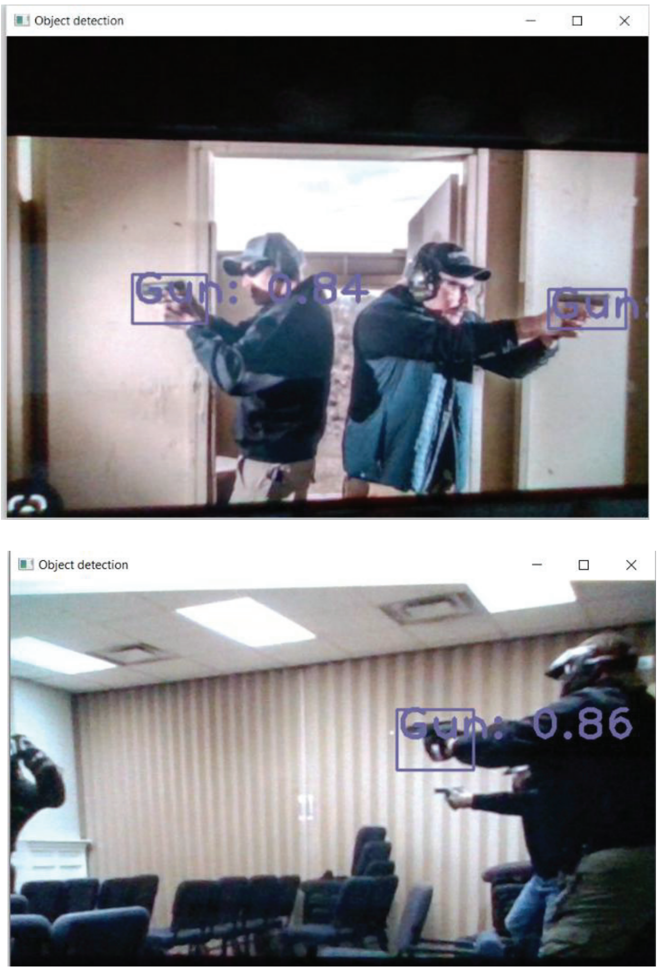


Fig. 8. Results of Experiment

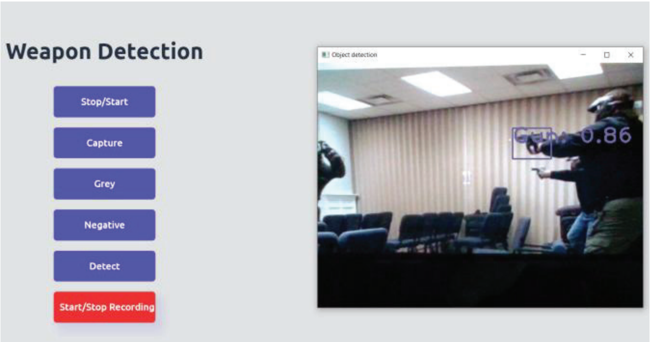


Fig. 9. System User Interface

V. Conclusion and Future Work

This research paper analysed the performance of four different detection models, namely YOLOv7, YOLOv8, CNN, and VGG, in detecting guns from a real time video surveillance. Overall, the YOLOv8 model is recommended as the best detection model for the dataset used in this study, considering its better overall performance and accuracy. The findings of this research paper can help in selecting the appropriate detection model for similar video surveillance applications.

As future work, the proposed weapon detection system can be further improved by adding parameters for differentiating between different types of guns and even many other weapons such as knives, bats etc. YOLO performance increases with the size of the dataset used for training. Multiple object overlapping may also be taken into consideration.

REFERENCES

[1] Sathyajit Loganathan, Gayashan Kariyawasam, Prasanna Sumathipala, "Suspicious Activity Detection in Surveillance Footage", International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019

[2] Harsh Jain, Aditya Vikram, Mohana, Ankit Kashyap, Ayush Jain, "Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications", International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), 2020

[3] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot Multibox Detector", Springer International Publishing AG, 2016

[5] Alberto Castillo , Siham Tabik , Francisco Pérez , Roberto Olmos , "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning", Francisco Herrera Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, Granada 18071, Spain

[6] Sarita Chaudhary, Mohd Aamir Khan, Charul Bhatnagar, "Multiple Anomalous Activity Detection in Videos", aGLA University, Mathura, 281406, India

[7] Peiyuan Jiang, Daji Ergu*, Fangyao Liu, Ying Cai, Bo Ma, "A Review of Yolo Algorithm Developments", Key Laboratory of Electronic and Information Engineering (Southwest Minzu University), Chengdu, 610049, China(ITQM 2020 & 2011)

[8] Arif Warsi , Munaisyah Abdullah , Mohd Nizam Husen , Muhammad Yahya, Sheroz Khan , Nasreen Jawaid ,"Gun detection system using YOLOv3",IEEE 6th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA 2019) 27-29 August 2019, Kuala Lumpur, Malaysia

[9] Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[10] Dr. N. Geetha, Akash Kumar. K. S, Akshita. B. P, Arjun. M,"Weapon Detection in Surveillance System", Coimbatore Institute of Technology