**ARM DETECTION IN SURVEILLANCE VIDEOS BY USING DEEP LEARNING ALGORITHMS**



**Abstract**

**Submitted by**

**N MANASA VENKATA NAGA SAI (193N1A0509)**

**K PRAMOD SAI (193N1A0524)**

**T SAI DARSHAN (193N1A0552)**

**S RAGHAVENDRA (193N1A0551)**

1. **KEERTHI (19711A0509)**

**Under the Guidance of**

1. **REKHA, M.TECH.,**

**Assistant Processor.**

**Department of Computer Science and Engineering**

**PVKK INSTITUTE OF TECHNOLOGY**

**Approved by AICTE, New Delhi, Affiliated to JNTUA, Ananthapuramu.**

**(Established under Sri Balaji Educational Society, Ananthapuramu).**

**ARM DETECTION IN SURVEILLANCE VIDEOS BY USING DEEP LEARNING ALGORITHMS**

**ABSTARCT**

This paper presents the development and optimization of a weapon detection system using deep learning techniques in surveillance videos. The system is based on a modified version of the You Only Look Once (YOLO) algorithm, which incorporates transfer learning and data augmentation to improve the detection accuracy. Furthermore, a novel loss function is proposed to balance the detection performance between different weapon categories. The proposed system outperforms the state-ofthe-art methods in terms of both accuracy and speed. The experimental results demonstrate the feasibility and effectiveness of the proposed system for weapon detection in real-world scenarios.

# Signature of the Supervisor

***Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications***

***Abstract***- Security is always a main concern in every domain, due to a rise in crime rate in a crowded event or suspicious lonely areas. Abnormal detection and monitoring have major applications of computer vision to tackle various problems. Due to growing demand in the protection of safety, security and personal properties, needs and deployment of video surveillance systems can recognize and interpret the scene and anomaly events play a vital role in intelligence monitoring. This paper implements automatic gun (or) weapon detection using a convolution neural network (CNN) based SSD and Faster RCNN algorithms. Proposed implementation uses two types of datasets. One dataset, which had pre-labelled images and the other one is a set of images, which were labelled manually. Results are tabulated, both algorithms achieve good accuracy, but their application in real situations can be based on the trade-off between speed and accuracy.

Keywords— Computer vision, weapon detection, Faster RCNN, SSD, CCTV,

Artificial Intelligence (AI).

# 1. INTRODUCTION

Weapon or Anamoly detection is the identification of irregular, unexpected, unpredictable, unusual events or items, which is not considered as a normally occurring event or a regular item in a pattern or items present in a dataset and thus different from existing patterns. An anomaly is a pattern that occurs differently from a set of standard patterns. Therefore, anomalies depend on the phenomenon of interest [3] [4]. Object detection uses feature extraction and learning algorithms or models to recognize instances of various category of objects [6]. Proposed implementation focuses on accurate gun detection and classification. Also concerned with accuracy, since a false alarm could result in adverse responses [11] [12]. Choosing the right approach required to make a proper trade-off between accuracy and speed. Figure 1 shows the methodology of weapons detection using deep learning. Frames are extracted from the input video. Frame differencing algorithm is applied and bounding box created before the detection of object.

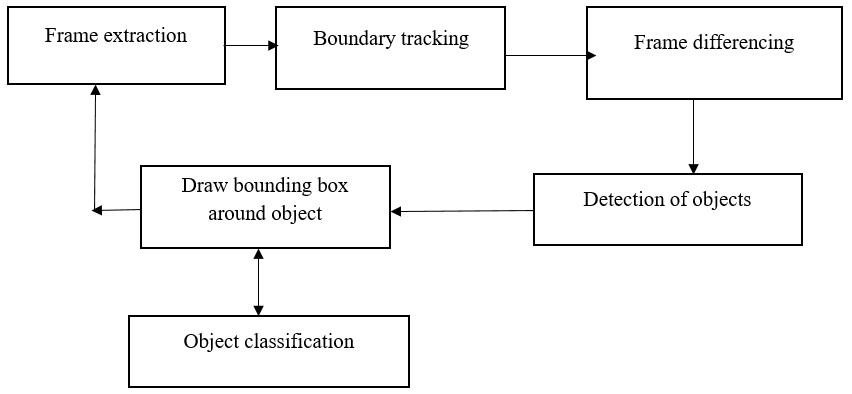


Fig.1.Methodology

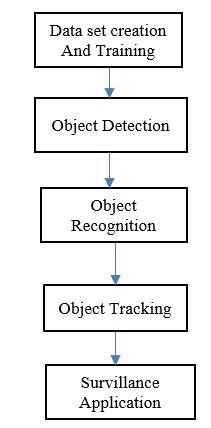


Fig.2. Detection and Tracking

# 2. IMPLEMENTATION

The flow of object detection and tracking as shown in figure 2. Dataset is created, trained and fed to object detection algorithm. Based on application suitable detection algorithm (SSD or fast RCNN) chosen for gun detection. The approach addresses a problem of detection using various machine learning models like Region Convolutional Neural Network (RCNN), Single Shot Detection (SSD) [2][9][15].

## A. Resources or components used for implementation

* *OpenCV 3.4- Open source computer vision library version 3.4.*
* *Python 3.5-**High level programming language used for various imageprocessing applications.*
* *COCO Dataset- Dataset consisting of common objects with respective labels.* 
  + *Anaconda**and Tensor flow 1.1*
  + *NVIDIA GeForce 820M GPU****-****GeForce is a brand of graphics processing units designed by Nvidia.*

## B. Dataset Specifications Case – I: Video specifications

* *System Configuration- Intel i5 7th Generation (4 Cores)*
* *Clock Speed- 2.5 GHz*
* *GPU- NVIDIA GeForce 820M*
* *Input Frames per Second- 29.97 fps*
* *Output Frames per Second- 0.20 fps*
* *Video Format- .mov*
* *Video Size- 4.14 MB*
* *COCO and self-created image dataset*

• *Number of classes trained- 5 Case –*

## Case – 2: Image specifications

* System Configuration- Intel i5 7th Generation (4 Cores)
* Clock Speed- 2.5 GHz
* GPU- NVIDIA GeForce 820M
* Input Image Size- 200-300 KB
* Training Time- ~0.6 seconds(SSD)
* ~1.7 seconds(RCNN)
* Image Format - .JPG
* COCO and self-created image dataset
* Number of classes trained for- 5

## C. Assumptions and Constraints made for implementation

* The gun is in line of sight of camera and fully/partially exposed to the camera.
* There is enough background light to detect the ammunition.
* GPU with high-end computation power were used to remove lag in the ammunition detection.
* This is not a completely automated system. Every gun detection warning will be verified by a person in charge.

## D. Faster R-CNN

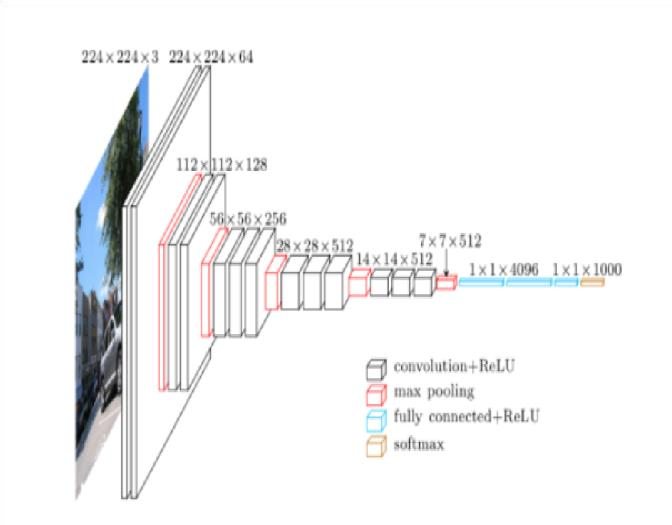


Fig 3. Layers in CNN Architecture [5]

Layers of CNN and faster RCNN architecture depicted in figure 3 and 4 respectively. It has two networks RPN to generate region proposals and network for object detection. To generate region proposals it uses selective search approach. Anchors or region boxes are ranked by RPN network.

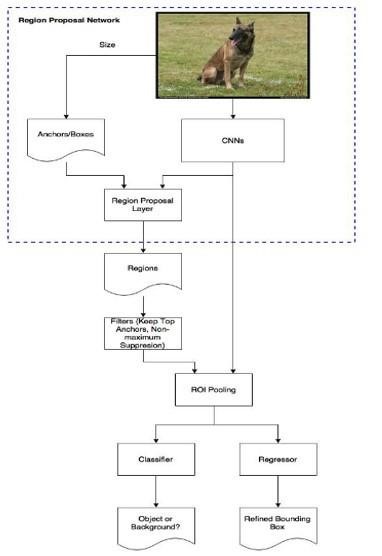


Fig 4. Faster R-CNN [5]

## Dataset Creation and Training

Images are downloaded in bulk using Fatkun Batch Image Downloader (chrome extension) which can download multiple Google Images at once. Then the downloaded images are labelled. 80% of total images used for training and 20% images for testing. The created ammunition dataset was then trained using Single Shot Detector (SSD) model and made 2669 iterations/steps on the model to ensure that the loss is less than 0.05 in order to increase the accuracy and precision. Figure 5 shows folder with test and train images. Figure 6 shows image with labels.

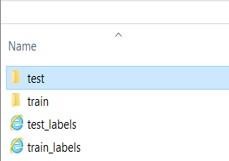


Fig.5. Folder with test and train images

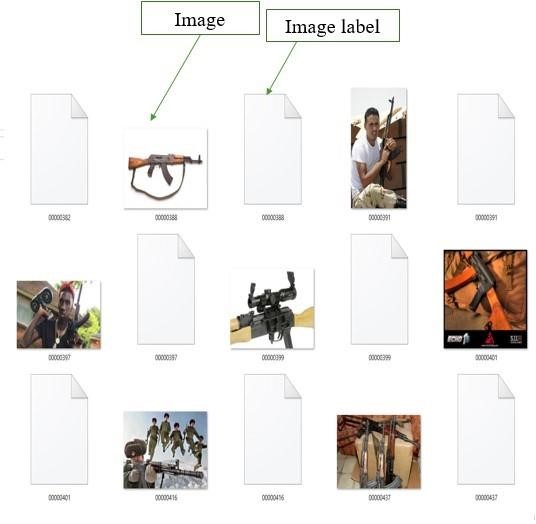


Fig.6. Image along with its label

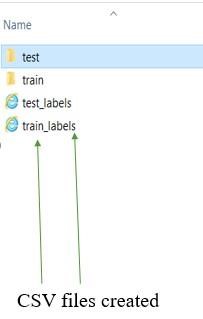
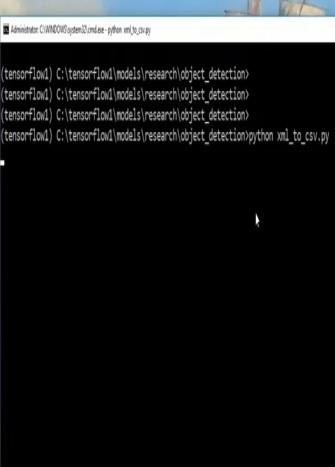


Fig.7. Command to create CSV files for the image labels

XML data is converted into CSV file by executing this command in Anaconda Prompt: pythonxml\_to\_csv.py as shown in figure 7

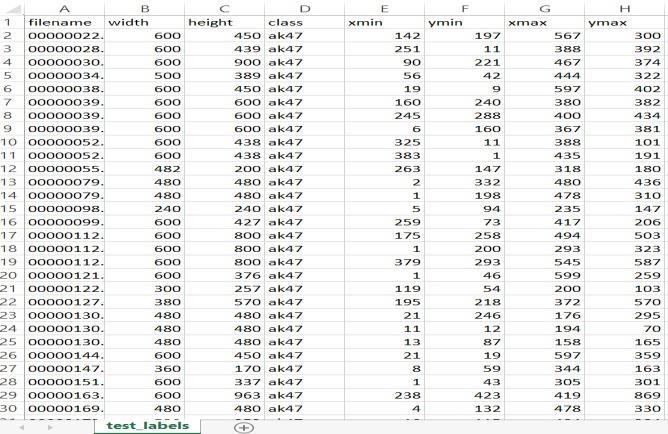
. 

Fig.8. CSV file of testing dataset

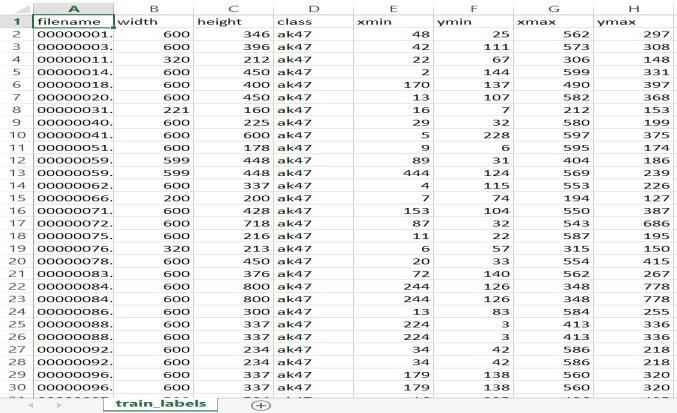


Fig.9. CSV file of Training dataset Figure 8 and 9

shows generated CSV file of test and training dataset. Figure10 shows the training of Faster R-CNN algorithm with training loss less than 0.15.Figure 11

describes pseudo code of faster RCNN.

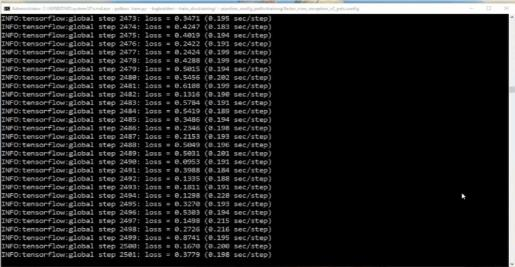


Fig.10.Training of Faster R-CNN

## Pseudo code of faster RCNN

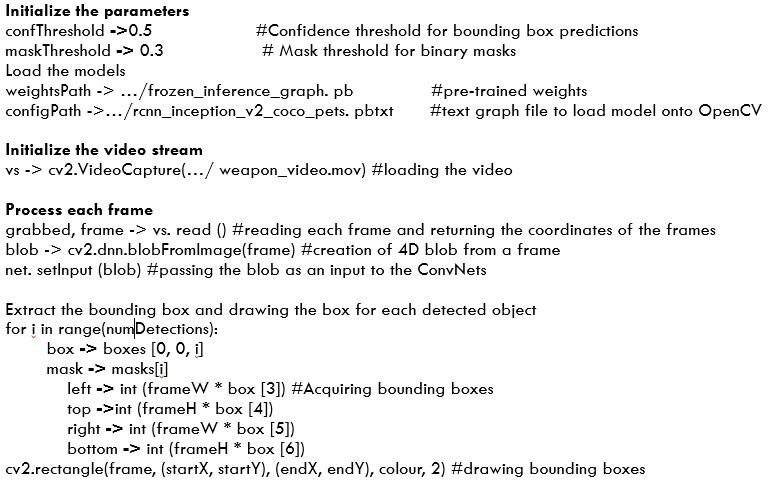


Fig.11. Pseudocode of faster RCNN

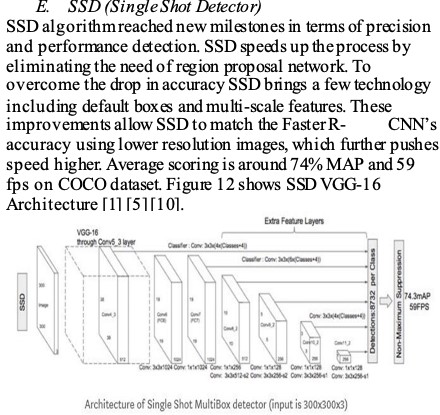


Fig.12. SSD VGG-16 Architecture [10]

Fig.14.Label map file to record all the names of classes

Fig.15. Command used to run SSD Model

SSD

until

trained

is

model

Fig.16.

0.05

to

reduced

is

loss

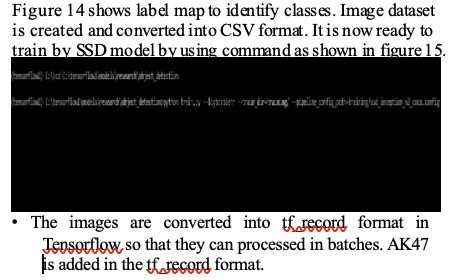


Fig.13. Manually labelled images



Figure 13 shows manually labelled images in xml format.



Total 838 considered for training and 241 images for testing



(22



% testing and 78% training).XML data is converted into



CSV file by executing this command in Anaconda Prompt:

-



python xml\_to\_csv.py.



Ground truth box and scale is



represented by



in



figure16.



Corresponding



pseudo



code



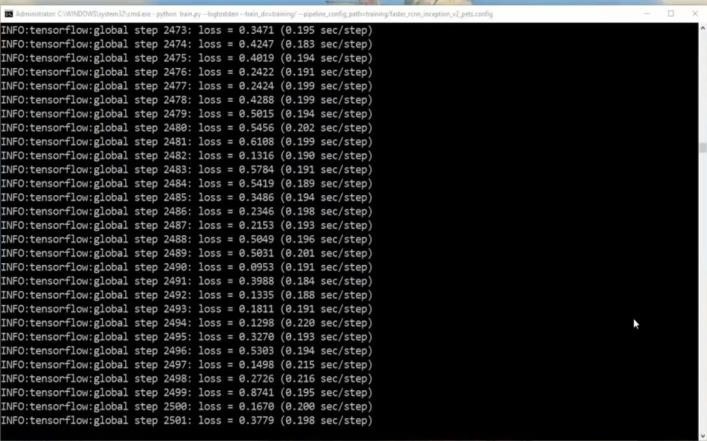
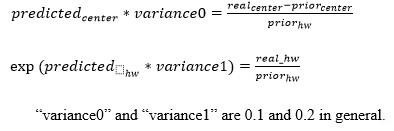
of



SSD



implementation as shown in figure 17.



***Pseudo code of SSD***Some of the drawbacks of faster RCNN compared to SSD model are training data is unwieldy and too long. SSD algorithm is a faster option, training occurs in more phases. Network is too slow at inference time and cannot provide accurate real time detection due to time spent on region proposals. SSD fills these shortcomings.Object detection and recognition To make sure object is detected, changes are made in the label map and tf\_record file. Label map is the file which stores the total number of types of objects that will be detected.

## 4. RESULTS AND ANALYSIS

### A. Detection of weapons using SSD algorithm Case 1: Using pre-labelled dataset

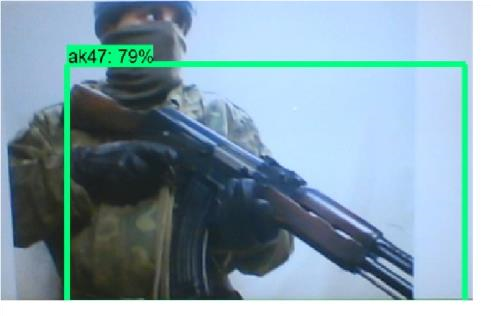


Fig.18. Detection of AK47 gun

Figure 18. Shows detection of a gun AK47 using SSD algorithm. Accuracy of detection is 79%. Further accuracy can be increased by increasing more number training samples.

***Case 2: Using self-created dataset***



Fig.19. COLT M1911 Detected with 72% accuracy



Fig.20. Smith & Wesson Model Detected with 67% accuracy

Figure 19 and 20 shows detection of COLT M1911 and Smith & Wesson Model gun with an accuracy of 72% and 67%.

### B. Detection of weapons using Faster R-CNN Case 1: Using pre-labelled image dataset



Fig.21. Detection of AK47 using Faster R-CNN

Figure 21 shows detection of AK47 gun using faster RCNN with an accuracy of 99% which shows Faster R-CNN provides the superior accuracy.



Fig.22. Detection of AK47 using Faster R-CNN

Figure 22 shows detection of AK47 in hands of army with accuracy of 99% and 81%.

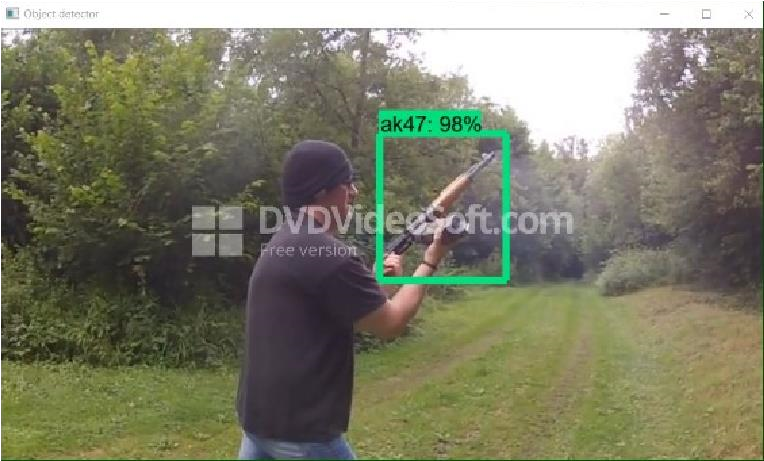


Fig.23. AK47 gun detection from video stream

Figure 23 shows detection of AK47 gun from video stream with an accuracy of 98%.

### Case 2: Using self-created image dataset



Fig.24. Detection of Colt M1911 gun

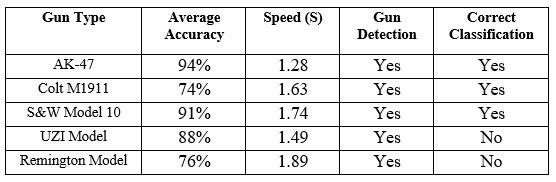


Fig.25. Detection of Smith & Wesson Model 10 gun

Figure 24 and 25 shows detection of Colt M1911 gun Smith & Wesson Model 10 gun using Faster R-CNN algorithm with the accuracy of 74% and 91%.

### 3. Performance Analysis Faster R-CNN

TABLE I. PERFORMANCE ANALYSIS: FASTER R-CNN ALGORITHM



From Table 1, it shows that highest average accuracy is obtained for pre-labelled dataset (i.e. AK47)

and the Colt M1911, Smith & Wesson Model 10, UZI Model, Remington

Model obtained accuracy in the range of 76% to 91%. Faster R-CNN achieves an average Accuracy of 84.6% and average speed 1.606s/frame. This concludes that the pre labelled dataset provided better accuracy because it is trained for millions of images in comparison to the self-created dataset.

*SSD*

TABLE II. PERFORMANCE ANALYSIS: SSD ALGORITHM

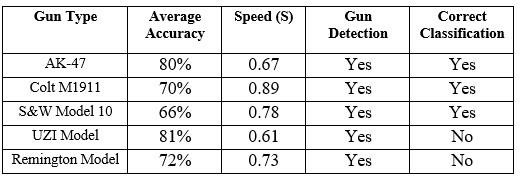


Table 2 shows performance Analysis for the SSD algorithm. Obtained SSD

Average Accuracy is 73.8% and average Speed

0.736 s/frame. Trained model for 5 classes of guns such as AK47, Smith and Wesson Model 10, Colt M1911, UZI Model and Remington model and obtained maximum confidence level for AK47 gun. Used SSD and RCNN Inception V2 models to train the guns. SSD took 12 more hours to train model in comparison with RCNN model but provided lower accuracy. Faster R-CNN achieved 10.8% more average accuracy than SSD algorithm. SSD provides faster speed than Faster R-CNN by 0.7 seconds. Pre-labeled dataset like AK47 gun provides higher accuracy in SSD and Faster R-CNN models, compared to self-created image dataset.

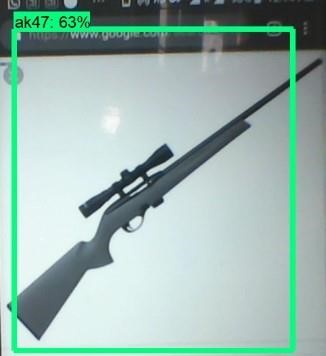


Fig.26 (A) UZI erroneously detected as AK 47 (b) Remington Model erroneously

detected as AK47

Figure 26 shows UZI and Remington model gun detection the detection for UZI Model gun and Remington Model gun is successful but the classification is incorrect. These guns are detected as AK47. Further Edge analytics concepts may be applied [13].

## 5. CONCLUSIONS

SSD and Faster RCNN algorithms are simulated for pre labeled and self-created image dataset for weapon (gun) detection. Both the algorithms are efficient and give good results but their application in real time is based on a tradeoff between speed and accuracy. In terms of speed,

SSD algorithm gives better speed with 0.736 s/frame. Whereas Faster RCNN gives speed 1.606s/frame, which is poor compared to SSD. With respect to accuracy, Faster RCNN gives better accuracy of 84.6%. Whereas SSD gives an accuracy of 73.8%, which is poor compared to faster RCNN.SSD provided real time detection due to faster speed but Faster RCNN provided superior accuracy. Further, it can be implemented for larger datasets by training using GPUs and high-end DSP and FPGA kits [16] [17].

## 6. REFERENCES

* Wei Liu et al., “SSD: Single Shot MultiBox Detector”, European Conference on Conputer Vision, Volume 169, pp 20-31 Sep. 2017.
* D. Erhan et al., “Scalable Object Detection Using Deep Neural Networks,”

*IEEE Conference on Computer Vision and Pattern Recognition*(CVPR),2014.

* Ruben J Franklin et.al., “Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks,” *International Conference on Inventive Systems and Control,2020.*
* H R Rohit et.al., “A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,”*2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud),* 2018.
* Abhiraj Biswas et. al., “Classification of Objects in Video Records using Neural Network Framework,” *International conference on Smart Systems and Inventive Technology,2018.*
* Pallavi Raj et. al.,“Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications” *IEEE International Conference on Intelligent Sustainable Systems,2019.*
* Mohana et.al., “Simulation of Object Detection Algorithms for Video Survillance Applications”, *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud),2018.*
* Yojan Chitkara et. al.,“Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model” *International Conference on Computing Methodologies and Communication,2019.*
* Rubner et.al, “A metric for distributions with applications to image databases”, International Conference on Computer Vision,201