iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37622

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Weapon and Object Detection Using Mobile-Net SSD Model in Deep Neural Network

S. Gopi Naik[1], M. Srinivasulu[2], K. Srikanth[3], N. Rama Devi[4], K. Mary Sudha Rani[5]
[1, 2, 3, 4, 5]*Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Osmania University*

Abstract: The plan is to establish an integrated system that can manage high-quality visual information and also detect weapons quickly and efficiently. It is obtained by integrating ARM-based computer vision and optimization algorithms with deep neural networks able to detect the presence of a threat. The whole system is connected to a Raspberry Pi module, which will capture live broadcasting and evaluate it using a deep convolutional neural network. Due to the intimate interaction between object identification and video and image analysis in real-time objects, By generating sophisticated ensembles that incorporate various low-level picture features with high-level information from object detection and scenario classifiers, their performance can quickly plateau. Deep learning models, which can learn semantic, high-level, deeper features, have been developed to overcome the issues that are present in optimization algorithms. It presents a review of deep learning based object detection frameworks that use Convolutional Neural Network layers for better understanding of object detection. The Mobile-Net SSD model behaves differently in network design, training methods, and optimization functions, among other things. The crime rate in suspicious areas has been reduced as a consequence of weapon detection. However, security is always a major concern in human life. The Raspberry Pi module, or computer vision, has been extensively used in the detection and monitoring of weapons. Due to the growing rate of human safety protection, privacy and the integration of live broadcasting systems which can detect and analyse images, suspicious areas are becoming indispensable in intelligence. This process uses a Mobile-Net SSD algorithm to achieve automatic weapons and object detection.

Keywords: Computer Vision, Weapon and Object Detection, Raspberry Pi Camera, RTSP, SMTP, Mobile-Net SSD, CNN, Artificial Intelligence.

I. INTRODUCTION

Weapon is the recognition of irregular, unexpected, unpredictable, or unusual events or objects that are not deemed to be an usually occurring event or a regular item in a pattern or items contained in a data-set, and hence different from current patterns. A pattern that is different from a set of typical designs is referred to as an anomaly. As a result, anomalies are dependent on the phenomenon of interest(phenomenon gives the multi-band and beta-band value due to the poor signal to noise ratio). Object detection recognises instances of several types of objects using feature extraction and learning techniques or models. The proposed implementation focuses on detecting and classifying weapons accurately. Also concerned about precision, as a false warning could result in negative consequences. To build an appropriate interchange between accuracy and rate, you'll need to choose the right address. The methodology for detecting weapons and objects based on the Mobile-Net SSD model. Before detecting the item, frames are collected from the input video streaming, a frame differencing technique is applied, and a bounding box is created. Raspberry Pi camera mounted at one place will capture all frames one by one and detect the objects into live footage and transmit data to the user using the Real-Time Streaming protocol (RTSP Server). RTSP server converts live video streaming server side to client side. After the detected frame it produces the bounding box with accuracy of each object in the scene. The G-Streamer framework can be used to play the live streaming video. To detect objects, the Raspberry Pi camera employs a pre-programmed deep learning model. When criminal acts are taking place in the area, this will record photographs and videos.

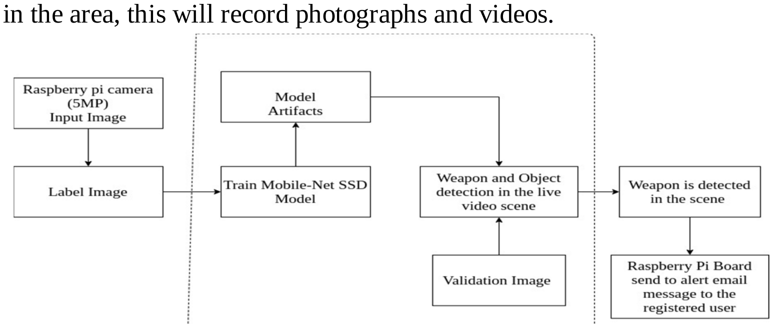


Figure 1. Weapon and Object Detection Block Diagram

II. DESCRIPTION OF BLOCK DIAGRAM

In this process (See figure 1), interpret an ARM processor-based raspberry pi board and a raspberry pi camera as an input image in this process, which is taken in real-time video streaming and detected by objects in the scene. If a person is holding a weapon, the camera will capture a photo and email it to the users who have registered. Users can respond quickly to apprehend the suspect before an accident occurs. The entire process is based on live broadcasting and datasets that have been trained to restore the model that was saved and send the picture to the restored model in order to get object detection for the number of images in the scene. Now, the Mobile-Net SSD model will yield the number of objects detected and their positions, also known as the region of interest. The Raspbian OS application will keep track of all communication paths throughout the system and send the appropriate commands to each interface.

III. PROBLEM DEFINITION

The main goal is to see if the person is carrying a weapon or not. If the person was brandishing a weapon, the Simple Mail Transfer Protocol sends an alert email message to the system or machine's operator. After that, the user can react by informing the cops, who will then proceed to apprehend the suspects before any accidents occur. In this process, we provide live video streaming via the RTSP communication protocol, as well as video recording and image capture for this operation.

IV. PROPOSED SYSTEM

In order to achieve basic functionalities like live broadcasting, image capturing and video recording. The Raspberry Pi module is used to achieve all the above basic functionalities. The RTSP communication protocol is used for live broadcasting. The Mobile-Net SSD model is used for weapon and object detection in a deep convolutional neural network.

V. FLOW CHART OF DETECTION TRACKING

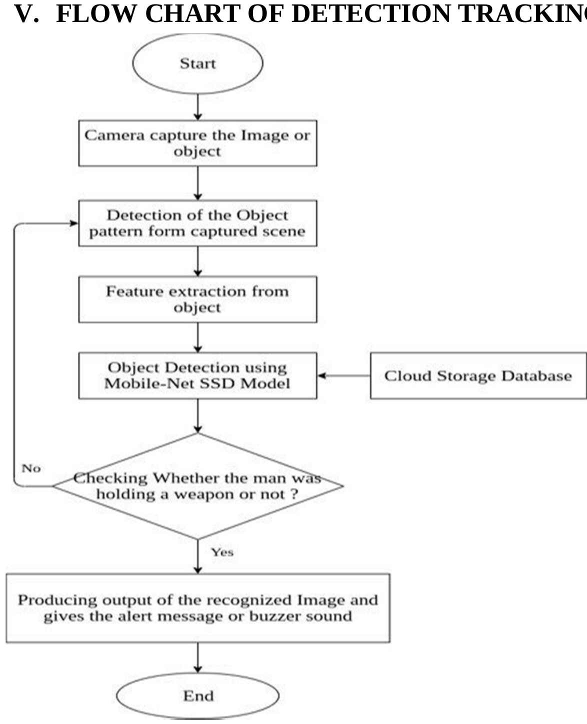


Figure 2. Flow of Weapon and Object Detection

Figure 2 illustrates that when the system starts, the camera will capture an image or video, then do feature extraction from the object, then apply the mobile-net SSD model to compress the image resolution to accelerate the image from cloud storage database or predefined data-set, and finally, if this step is completed, our model will check whether the person was holding a weasel. Users can react to alert the cops, and cops will respond to apprehend the suspects before any accidents occur.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

VI. IMPLEMENTATION

A. Resource Components used for Implementation

1) OpenCV 4.5.3- The library is cross-platform and free to use under the open-source BSD license.
2) Python 3.9- High level programming language used for various image processing applications.
3) Cloud datasets or self created images- Dataset consisting of common objects with respective labels.
4) Tensorflow 2.5.0, pillow, H5py, Matplotlib and Scipy.
5) NVIDIA GeForce 820M GPU-GeForce is a brand of graphics processing units designed by NVIDIA.

B. Datasets Specification

1) *Step-1: Video Specification*
   a) Operating system: Raspberry pi OS, FreeBSD.
   b) System on a chip: Pi 3B+, Broadcom BCM2837 64 bit CPU (Quad Core 1.2GHz), 1GB RAM.
   c) BCM43438 wireless LAN and Bluetooth Low Energy on board.
   d) CSI camera port for connecting a raspberry pi camera, full size HDMI port and 100 Base Ethernet.
   e) Clock Speed - 2.5GHz to 5GHz.
   f) Input Frame per seconds - 0.17 to 0.23 sec/frame
   g) Output Frame per seconds - 0.59 sec/frame
   h) Video format - .Mp4
   i) Video size - 5MB
   j) COCO and self-created images or real time images.
2) *Step-2: Image Specification*
   a) System on a chip: Pi 3B+, Broadcom BCM2837 64 bit CPU (Quad Core 1.2GHz), 1GB RAM.
   b) CSI camera port for connecting a raspberry pi camera and trained datasets.
   c) Clock speed - 2.5GHz to 5GHz.
   d) GPU-NVIDIA GeForce 820M.
   e) Input image size- 300 to 400 KB.
   f) Training time - ~0.6 sec(Mobile-Net SSD).
   g) Image format- .jpg or .png.

C. Assumptions and Constraints made for Implementation

1) The weapon is in the scene of the camera and fully or partially exposed to the camera.
2) There is no background light to detect the ammunition.
3) GPUs with high end computation power were used to remove lag in the ammunition detection.
4) This is not a completely automated system. Every weapon detection warning will be verified by a user in the scene.

D. Convolution Neural Network (CNN)

A Convolution Neural Network is a type of artificial neural network used in image recognition and it detect the image with in the live video and divide the frames to the image into cells with assigning anchor box, Anchor Box means set of predefined boundary box of a certain calculation of height and width. See figure 3. illustrates the three layers configuration of CNN that is input layer, hidden layer and output layer. To generate region proposals it uses a selective search method. Anchor or region boxes are ranked by CNN network. CNN's are strong in image processing and video recognition that use Deep Learning to perform both generative and descriptive tasks, often using computer vision that include image and video recognition along with recommended system and natural language processing like python, java etc.

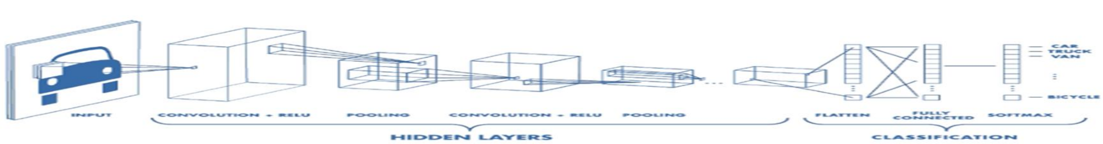


Figure 3. Architecture of a CNN

VII. MOBILE-NET SSD MODEL AS A METHODOLOGY

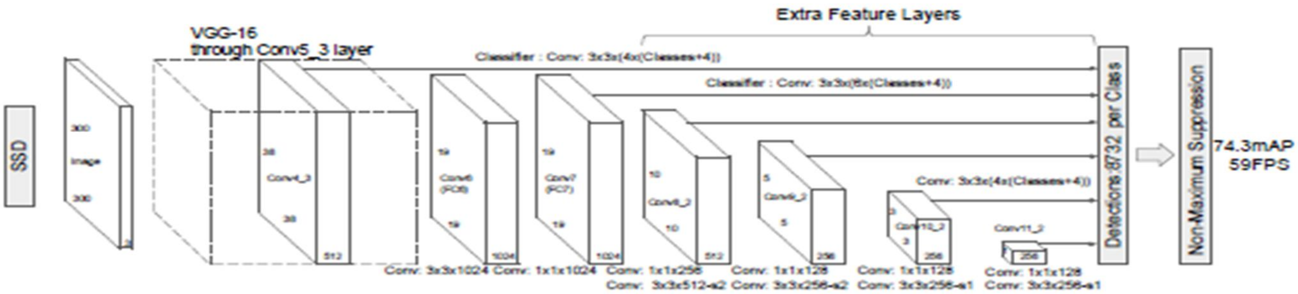


Figure 4. Architecture of a Mobile-Net SSD VGG-16 Model

Mobile-Net is a light-weight deep neural network. The SSD technique is based on a sustain of neural network that makes for a fixed-size cluster of leap boxes and scores for the existence of object category instances in leap boxes. Above figure 4. Illustrates the Single Shot Multibox Detector takes one single shot to detect multiple objects within the images. SSD is designed to be an individual of the base network. In object detection, we classify an image and recognize where an object occupies an image. For this we have to obtain the bounding box, frame detection is considered a regression problem. SSD defines a scale value for each feature map layer. Starting from the left, Conv4_3 detects objects at the smallest scale 0.2 (or 0.1 sometimes), and then increases linearly to the rightmost layer at a scale of 0.9. Combining the scale value with the target aspect ratios, we compute the width and the height of the default boxes. For layers making 6 predictions, SSD starts with 5 target aspect ratios: 1, 2, 3, 1/2, and 1/3.
The proposed methodology uses Mobile-Net SSD algorithm with caffe for detecting various real time objects.
It has two composed parts:

1) Extract feature maps.
2) Apply convolutional filters to detect objects.

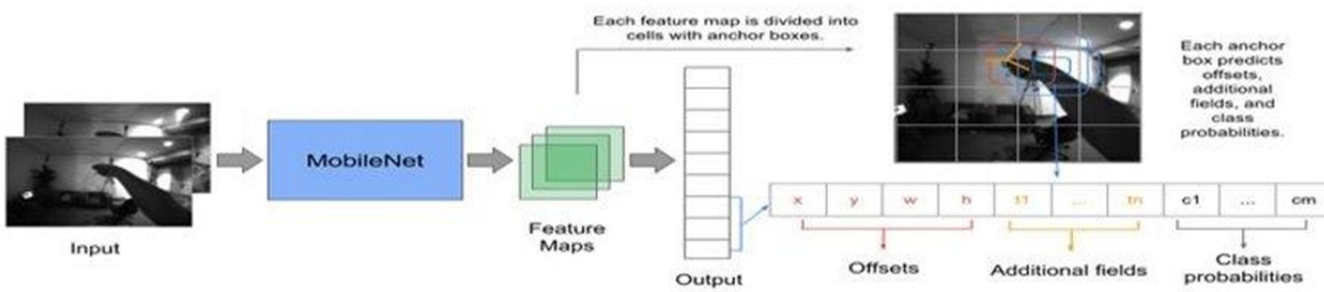


Figure 5. Object Detection Frame

Above figure 5. Illustrates whenever the camera captures the image or video, its exact feature map is divided into cells with anchor boxes, after that apply convolutional filters to detect objects, each anchor box predicts offsets, additional fields and class probabilities.

A. SSD has two Loss Functions

1) *Localization Loss:* It calculates how far aside the network's forecast bounding boxes are from the floor variety.

The localization loss between the predicted box l and the ground truth box g is defined as the smooth L1 loss with cx, cy as the offset to the default bounding box d of width w and height h.

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \mathrm{smooth}_{L1}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right)$$

$$x_{ij}^{p} = \begin{cases} 1 & \text{if } IoU > 0.5 \text{ between default box } i \text{ and ground true box } j \text{ on class } p \\ 0 & \text{otherwise} \end{cases}$$

2) *Confidence Loss:* It calculates how confident the network is of the targeting of the computed bounding boxes.

It is calculated as the softmax loss over multiple classes confidences c (class score).

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

where N is the number of matched default boxes.

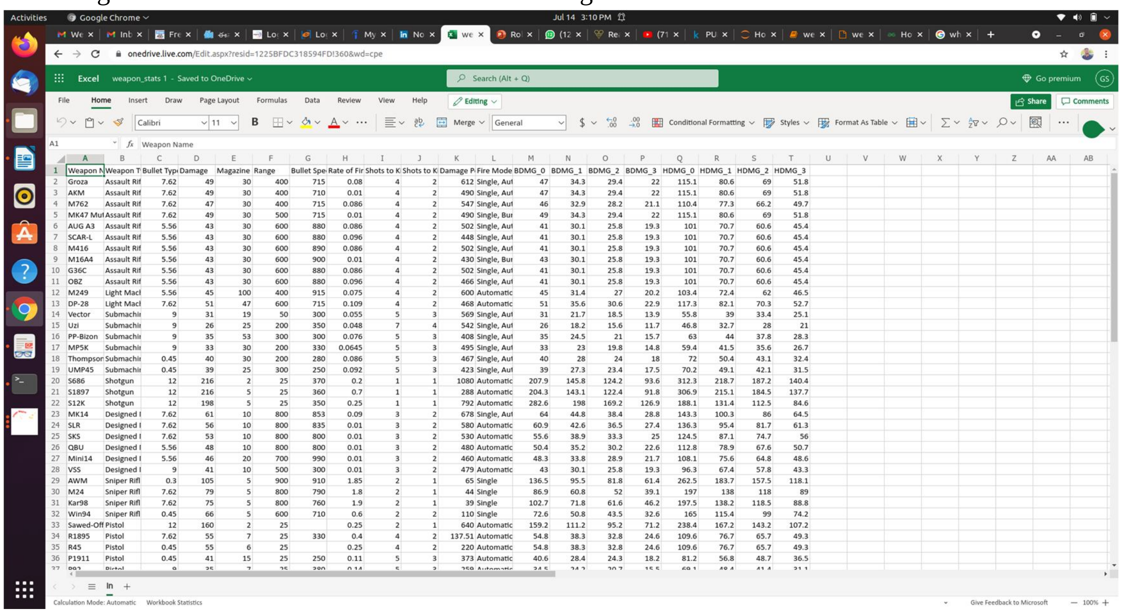
The final loss function is computed as:

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

Where N is the number of positive matches and α is the weight for the localization loss.

B. Training of Mobile-Net SSD Model

Below figure 6. It illustrates the manually labelled images in.CSV file format. Datasets were collected from Google Images and normally captured images with a camera in different situations. The images consist of a person holding a weapon or not holding a weapon in different conditions, and it detects some common objects like weapons, knives, cars, motorcycles, aeroplanes, chairs, people, pens, etc. The trained data is 80% and for testing we have used only 20% of the data. datasets were sourced from kaggle images and manually captured images and video. The total number of images is 500, collected under various situations. The training set consists of 400 images and the validation set consists of 100 images. The total size of the datasets is 250 MB.



Weapon N	Weapon T	Bullet Typ	Damage	Magazine	Range	Bullet Spe	Rate of Fir	Shots to K	Shots to K	Damage P	Fire Mode	BDMG_0	BDMG_1	BDMG_2	BDMG_3	HDMG_0	HDMG_1	HDMG_2	HDMG_3
Groza	Assault Rif	7.62	49	30	400	715	0.08	4	2	612	Single, Aut	47	34.3	29.4	22	115.1	80.6	69	51.8
AKM	Assault Rif	7.62	49	30	400	710	0.01	4	2	490	Single, Aut	47	34.3	29.4	22	115.1	80.6	69	51.8
M762	Assault Rif	7.62	47	30	400	715	0.086	4	2	547	Single, Aut	46	32.9	28.2	21.1	110.4	77.3	66.2	49.7
MK47 Mut	Assault Rif	7.62	49	30	500	715	0.01	4	2	490	Single, Bur	49	34.3	29.4	22	115.1	80.6	69	51.8
AUG A3	Assault Rif	5.56	43	30	600	880	0.086	4	2	502	Single, Aut	41	30.1	25.8	19.3	101	70.7	60.6	45.4
SCAR-L	Assault Rif	5.56	43	30	600	880	0.096	4	2	448	Single, Aut	41	30.1	25.8	19.3	101	70.7	60.6	45.4
M416	Assault Rif	5.56	43	30	600	890	0.086	4	2	502	Single, Aut	41	30.1	25.8	19.3	101	70.7	60.6	45.4
M16A4	Assault Rif	5.56	43	30	600	900	0.01	4	2	430	Single, Bur	43	30.1	25.8	19.3	101	70.7	60.6	45.4
G36C	Assault Rif	5.56	43	30	600	880	0.086	4	2	502	Single, Aut	41	30.1	25.8	19.3	101	70.7	60.6	45.4
OBZ	Assault Rif	5.56	43	30	600	880	0.096	4	2	466	Single, Aut	41	30.1	25.8	19.3	101	70.7	60.6	45.4
M249	Light Macl	5.56	45	100	400	915	0.075	4	2	600	Automatic	45	31.4	27	20.2	103.4	72.4	62	46.5
DP-28	Light Macl	7.62	51	47	600	715	0.109	4	2	468	Automatic	51	35.6	30.6	22.9	117.3	82.1	70.3	52.7
Vector	Submachi	9	31	19	50	300	0.055	5	3	569	Single, Aut	31	21.7	18.5	13.9	55.8	39	33.4	25.1
Uzi	Submachi	9	26	25	200	350	0.048	7	4	542	Single, Aut	26	18.2	15.6	11.7	46.8	32.7	28	21
PP-Bizon	Submachi	9	35	53	300	300	0.076	5	3	408	Single, Aut	35	24.5	21	15.7	63	44	37.8	28.3
MP5K	Submachi	9	33	30	200	330	0.0645	5	3	495	Single, Aut	33	23	19.8	14.8	59.4	41.5	35.6	26.7
Thompson	Submachi	0.45	40	30	200	280	0.086	5	3	467	Single, Aut	40	28	24	18	72	50.4	43.1	32.4
UMP45	Submachi	0.45	39	25	300	250	0.092	5	3	423	Single, Aut	39	27.3	23.4	17.5	70.2	49.1	42.1	31.5
S686	Shotgun	12	216	2	25	370	0.2	1	1	1080	Automatic	207.9	145.8	124.2	93.6	312.3	218.7	187.2	140.4
S1897	Shotgun	12	216	5	25	360	0.7	1	1	288	Automatic	204.3	143.1	122.4	91.8	306.9	215.1	184.5	137.7
S12K	Shotgun	12	198	5	25	350	0.25	1	1	792	Automatic	282.6	198	169.2	126.9	188.1	131.4	112.5	84.6
MK14	Designed I	7.62	61	10	800	853	0.09	3	2	678	Single, Aut	64	44.8	38.4	28.8	143.3	100.3	86	64.5
SLR	Designed I	7.62	56	10	800	835	0.01	3	2	580	Automatic	60.9	42.6	36.5	27.4	136.3	95.4	81.7	61.3
SKS	Designed I	7.62	53	10	800	800	0.01	3	2	530	Automatic	55.6	38.9	33.3	25	124.5	87.1	74.7	56
QBU	Designed I	5.56	48	10	800	800	0.01	3	2	480	Automatic	50.4	35.2	30.2	22.6	112.8	78.9	67.6	50.7
Mini14	Designed I	5.56	46	20	700	990	0.01	3	2	460	Automatic	48.3	33.8	28.9	21.7	108.1	75.6	64.8	48.6
VSS	Designed I	9	41	10	500	300	0.01	3	2	479	Automatic	43	30.1	25.8	19.3	96.3	67.4	57.8	43.3
AWM	Sniper Rifl	0.3	105	5	900	910	1.85	2	1	65	Single	136.5	95.5	81.8	61.4	262.5	183.7	157.5	118.1
M24	Sniper Rifl	7.62	79	5	800	790	1.8	2	1	44	Single	86.9	60.8	52	39.1	197	138	118	89
Kar98	Sniper Rifl	7.62	75	5	800	760	1.9	2	1	39	Single	102.7	71.8	61.6	46.2	197.5	138.2	118.5	88.8
Win94	Sniper Rifl	0.45	66	5	600	710	0.6	2	2	110	Single	72.6	50.8	43.5	32.6	165	115.4	99	74.2
Sawed-Off	Pistol	12	160	2	25		0.25	2	1	640	Automatic	159.2	111.2	95.2	71.2	238.4	167.2	143.2	107.2
R1895	Pistol	7.62	55	7	25	330	0.4	4	2	137.51	Automatic	54.8	38.3	32.8	24.6	109.6	76.7	65.7	49.3
R45	Pistol	0.45	55	6	25		0.25	4	2	220	Automatic	54.8	38.3	32.8	24.6	109.6	76.7	65.7	49.3
P1911	Pistol	0.45	41	15	25	250	0.11	5	3	373	Automatic	40.6	28.4	24.3	18.2	81.2	56.8	48.7	36.5

Figure 6. Manually Labelled Images in .CSV file

C. Pseudo code of Mobile-Net SSD Model

The 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.

1) *Initialize the Parameters*

Def -> user can register with their email for sending threats whenever the weapon is detected.
confThreshold -> 0.5
maskThreshold -> 0.3

2) *Load the Models*

Weightspath->net=cv2.dnn.readNetFromCaffe(args.prototxt, args.weights)
configpath->confidence = detections[0, 0, i, 2]

3) Initialize the Video Stream

vs->cv2.videocapture(.../real_time_source.mp4)

4) Process Each Frame

grabbed,frame-> vs.read()
blob->cv2.dnn.blobFromImage(frame)
net.setInput(blob)

5) Extract the Bounding Box and Drawing the Box For Each Detected Object

For i in range(numDetections):
box->boxes [0,0,i]
mask->mask[i]
left->int(frameW*box[3])
top->int(frameH*box[4])
right->int(frameW*box[5]
bottom->int(frameH*box[6])
cv2.rectangle(frame,(startX,startY),(endX,endY),color,2)

VIII. RESULTS AND DISCUSSIONS

A. Detection of Weapons Using Mobile-Net SSD Algorithm

Below figure 7 illustrates the system will detect the object from the grabbed image using the Mobile-Net SSD Model. The main goal is to identify the weapons. It checks whether the person was holding a weapon or not. If yes, the Raspberry Pi module will send a warning email message to registered users using SMTP (Simple Mail Transfer Protocol), and then the user will receive an email message. Users can give the information to cops, who will then apprehend the suspects before any accidents occur.

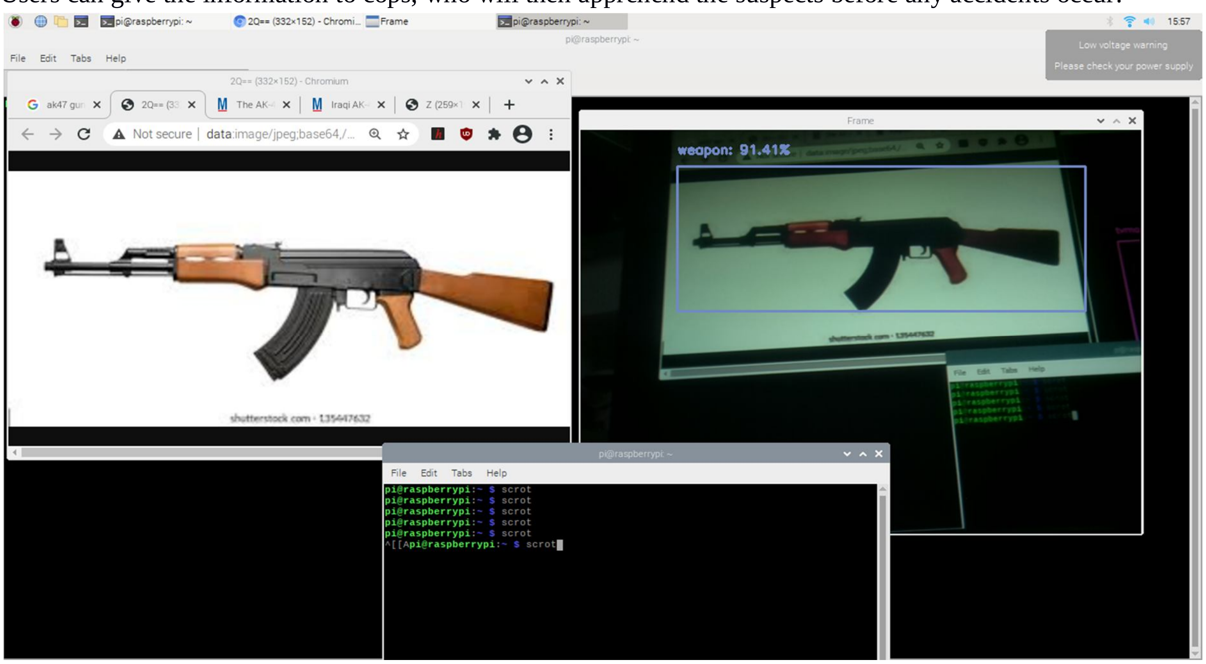


Figure 7. Weapon detected in the scene

As shown below in figure 8 A person is holding a weapon, the Raspberry Pi module will take the image and send an alert mail message using SMTP protocol to the registered user. Not only guns, the rest of the weapon also will detect the machine like missiles, knifes, blades etc. At same time the system will detect the objects in live broadcasting and trained images or manually labelled images or normally captured images.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Figure 8. Person holding a weapon

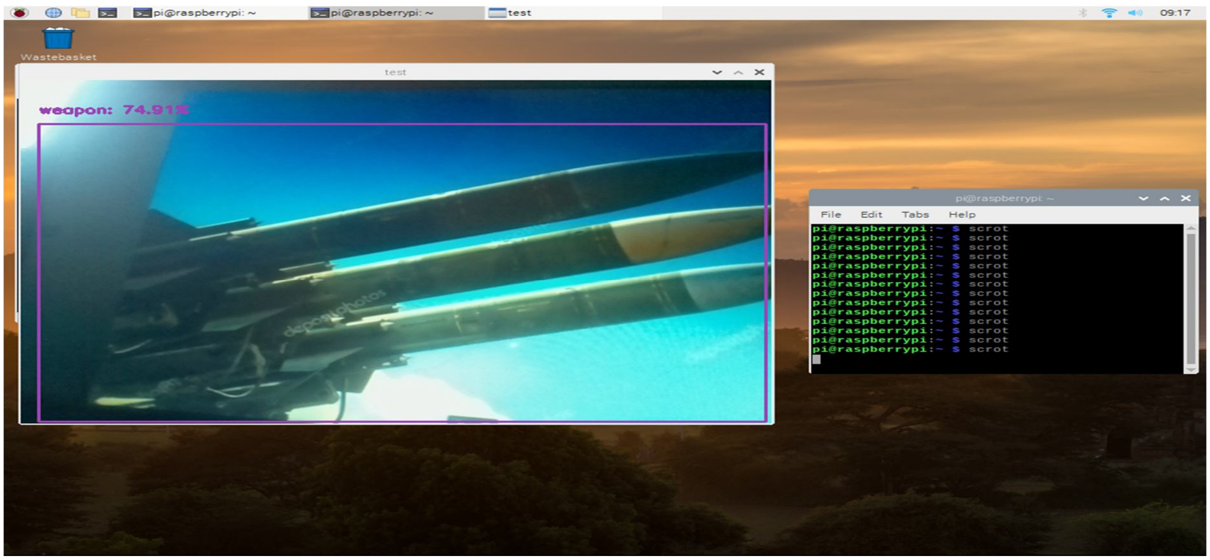


Figure 9 Detection of Missiles

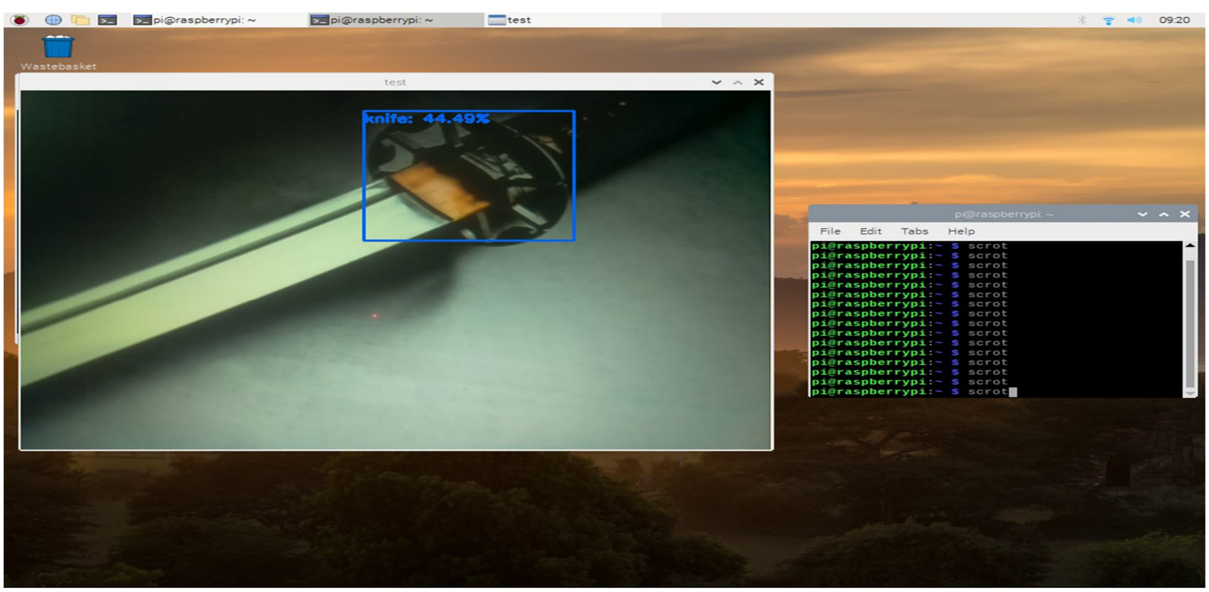


Figure 10 Detection of a Knife

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

B. Multiple Objects Detected in Single Shot

The system will detect in a single shot within the image what is inside the image first, compress the image, and offer the system will give the estimated accuracy of each object like Car- 89.59%, Wheel- 85.5%, Truck-83.21%, Person-95.26% and Motorbike-92.35% as illustrated in figures 11 and 12.

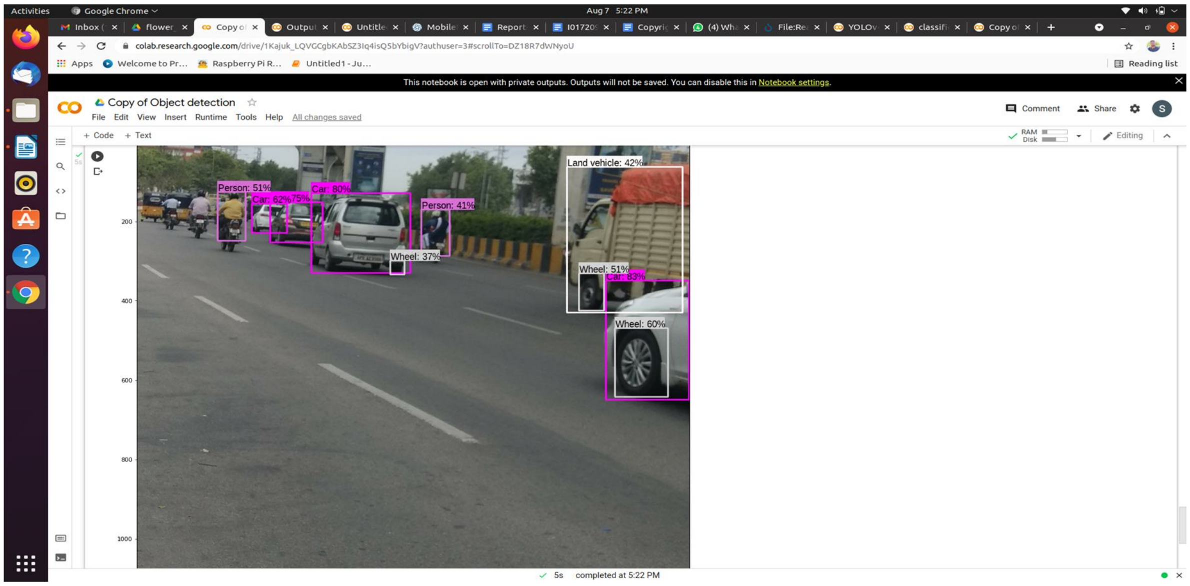


Figure 11 Detection of Multiple Objects in Single Shot

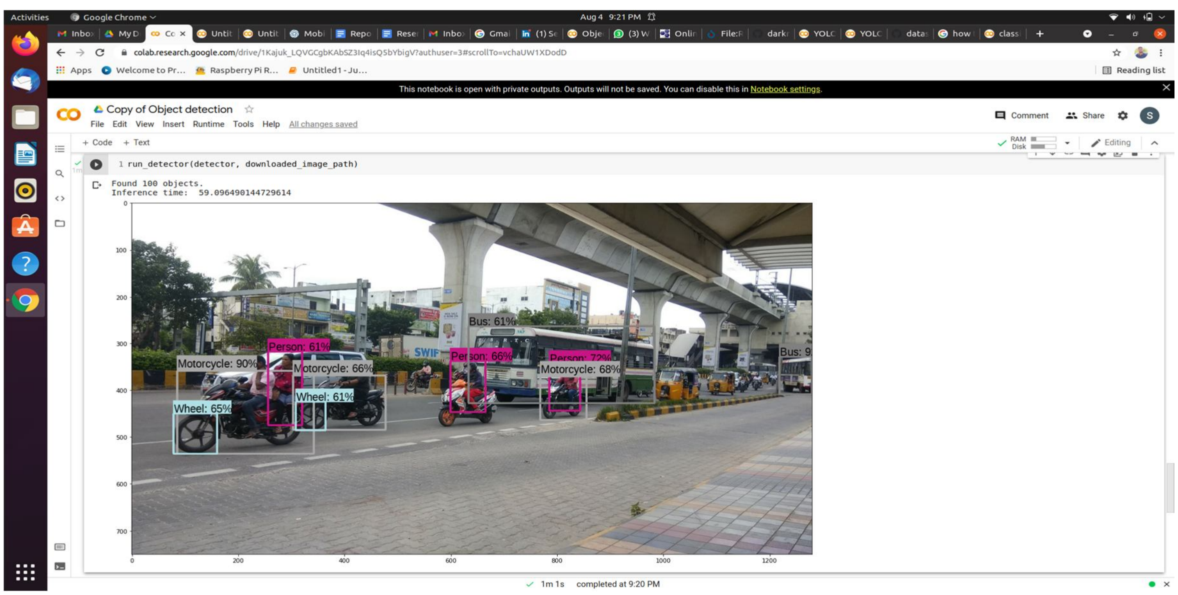


Figure 12 Detection of Multiple Objects in Single Shot

C. Training, Validation Accuracy, and Training, Validation Loss: Plotting Graph

When training and validating datasets to construct a model, the graph shows that the blue line represents training accuracy and the orange line represents dataset validation accuracy. When displaying the graph with different epochs, the process will lose functions, accuracy, validation loss, and validation accuracy. When running more epochs, the loss function decreases and the epochs grow. The Mobile-Net SSD model gets better at training datasets and validation datasets. The training and validation losses are reduced by running epochs and the Mobile-Net SSD model fits for each visualisation data and delivers improved accuracy for each dataset. The x-axis represents epochs and the y-axis represents the loss function, now compiling the entered epochs. Here, observe how the plotted line is increasing and decreasing in the graph.

A brief explanation of plotted lines, whenever training loss decreases as well as training accuracy increases, in the same way, validation loss will decrease as well as validation accuracy will increase, as shown below in figure 13.

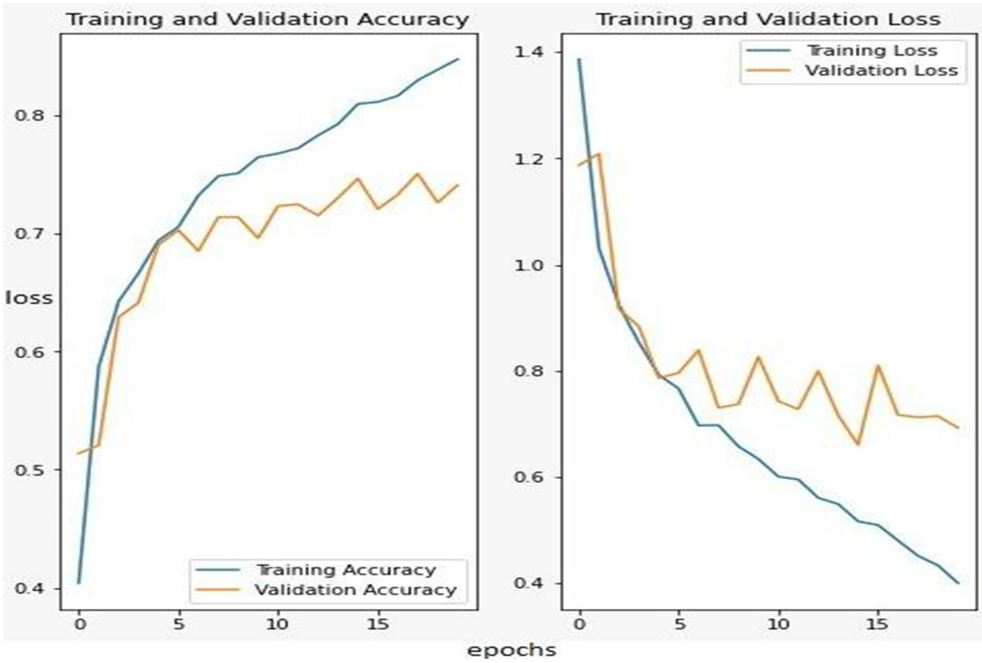


Figure 13 Training and Validation Accuracy

Tabel-1: Performance Analysis: Mobile-Net SSD Algorithm

Weapon or Image object	Average Accuracy	Speed	Weapon or Object Detection	Classification
Weapon or gun	91.25%	0.66	Yes	Yes
Missile	75.26%	0.59	Yes	Yes
Knife	72.69%	0.49	Yes	No
Person	95.26%	0.25	Yes	Yes
Car	92.14%	0.39	Yes	No
Motorbike	70.48%	0.53	Yes	No

IX. CONCLUSION AND FUTURE SCOPE

In everyday life, technology is essential to human existence. That's why, in this process, the machine achieved some basic functionalities, such as live broadcasting, object detection, and weapon detection, with the help of a raspberry pi module for detecting real-time objects. Whenever a weapon is detected on the scene, the SMTP (Simple Mail Transfer Protocol) server sends an alert email message to registered users. For weapon detection using predetermined algorithms in this project, users may simply identify the suspect before an accident occurs. The Mobile-Net SSD model for weapon and object detection is simulated using pre-labeled and self-created images. Although SSD approaches are economical and offer excellent results, they require a trade-off between speed and accuracy when used in real time. The SSD approach is faster in terms of performance, coming in at 0.173 seconds per frame. When compared to SSD, the faster RCNN has a frame rate of 1.606 seconds per frame, which is extremely slow. SSD has a precision of 75.3 percent on average, whereas YOLO has a precision of 74.5 percent and Faster RCNN has a precision of 73.2 percent on average.

A. Future Scope

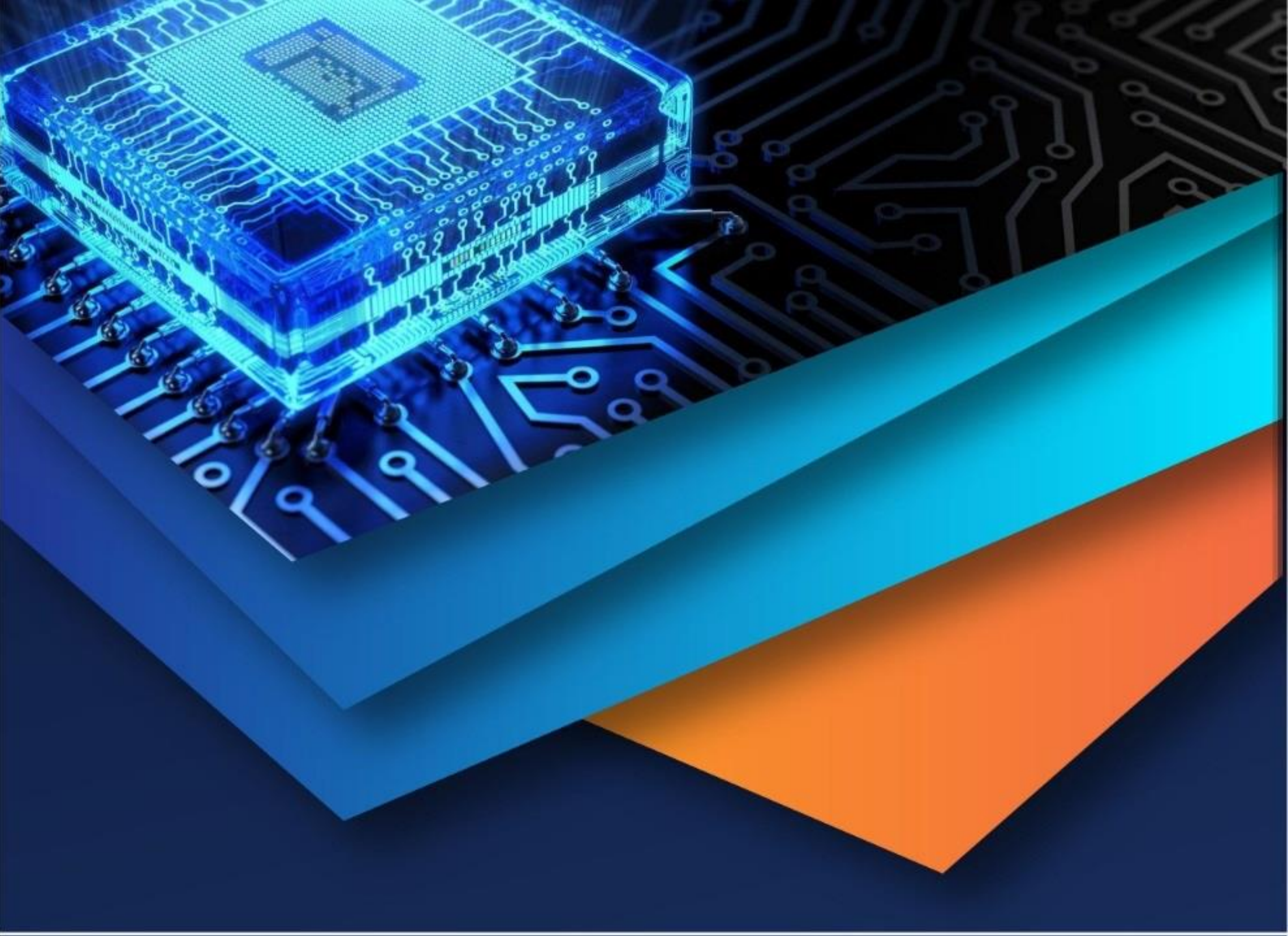
With the advent of techniques to add the Pan Tilt Zoom camera for 360-degree view and object detection, using an optical lens to zoom in on any object for better resolution. The detection of bombs and bullets. Identify underground water levels using the Raspberry Pi module (under the depth of 170m).

X. ACKNOWLEDGEMENT

While bringing out this project to its final form, I came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey my gratitude to all of them. First and foremost, I would like to express my deep sense of gratitude and indebtedness to my supervisor Smt. N. Rama Devi and Smt. K. Mary Sudha Rani for their valuable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. I am also thankful to the Head of the Department Dr. Y. Rama Devi for providing excellent infrastructure and such a nice atmosphere for completing this project successfully. Finally, I would like to take this opportunity to thank my family and friends for their support throughout this work. I also sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

REFERENCES

[1] Wei Liu et al., "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision, Volume 169, pp 20-31 Sep. 2017.
[2] D. Erhan et al., " Scalable Object Detection Using Deep Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2014.
[3] Ruben J Franklin et.al., "Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks," International Conference on Inventive Systems and Control,2020.
[4] 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.
[5] Abhiraj Biswas et. al., "Classification of Objects in Video Records using Neural Network Framework," International conference on Smart Systems and Inventive Technology,2018.
[6] 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.
[7] Mohana et.al., "Simulation of Object Detection Algorithms for Video Surveillance Applications", International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud),2018.
[8] Yojan Chitkara et. al.," Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model" International Conference on Computing Methodologies and Communication,2019.
[9] Rubner et.al, "A metric for distributions with applications to image databases", International Conference on Computer Vision,2016
[10] N. Jain et.al., "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2019.
[11] A. Glowacz et.al., "Visual Detection of Knives in Security Applications using Active Appearance Model",Multimedia Tools Applications,2015.



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY