Intelligent Rover: An IoT Based Smart Surveillance Robotic Car for Military

Safa Mohammed Sali
Electronics and Communication Engineering
Sahrdaya College of Engineering and Technology
Kodakara, India
safamohammadsalih@gmail.com

Dr. K.R. Joy
Electronics and Communication Engineering
Sahrdaya College of Engineering and Technology
Kodakara, India
krjoy@sahrdaya.ac.in

***Abstract*— Intelligent Rover is a military based surveillance robotic car that is trained to detect human beings and weapons while moving according to the user need. Intelligent Rover is controlled via smart phone and uses raspberry pi and Arduino to achieve the mission of the rover. It captures live video using picamera and streams it, which can be viewed from a webpage. The camera can be swiveled horizontally and vertically. Intelligent Rover employs yolo V3 algorithms to detect weapons and persons individually. When it detects any objects, it notifies the user via an email. The DC motors connected to the wheels sets the rover into motion that is controlled through the webpage from smartphone or any other devices. This robotic automobile is also capable of detecting metals via sensors and notifies the same to the user via email.**

Keywords*— *Raspberry pi, object detection, yolo V3, IoT, metal detection

I. Introduction

The internet of things (IoT) embeds software, sensor, processors and other devices and technologies that are utilized to link and share data and information from one device to the other with the help of the internet. The world of IoT does not have any extremities and extends to a wide range of applications that are advantageous for the mankind by predominantly safeguarding them and their properties.

The advent of IoT has succored humans in numerous areas spotlighting security in military field. There have been different systems and devices for the surveillance of the intruders, weapons and metals. But most of such devices are very complex in their design and are very expensive. [1] shows an IoT based smart sniper that is automated to detect enemy and extinguish them by firing manually. [2] presents a device that is efficient in capturing, extracting and processing the information existing in the war field and transferring these in the form of a caution or an alert to the troopers. [3] conveys a military based object detection algorithm using hyperspectral imagery (HSI).

In this paper, we present an intelligent surveillance robotic rover that operates on the principles of IoT. This robotic device is used for the surveillance in the military field with the potential to detect guns, people and metal, live stream these and notifying the user about the same through an email. This device assists the troopers in detecting the enemy and traps using yolo V3 algorithm and sensors, thereby saving their precious lives. The scene of detected object is saved for future reference. The predominant feature of the rover is that it can be controlled according to the user needs by steering the car left, right, forward and backwards via smartphone. The camera can be swiveled horizontally and vertically. The angle of tilting and panning can be varied according to the user. Fig. 1 depicts the features of intelligent rover as a block diagram.

We organize this paper in five sections. In section II, we discuss about the detection of guns and persons. In section III, we present the hardware and other necessary technologies for the detection of metals. Section IV and the succeeding sections outline the swiveling technique of the camera, movement of the rover and unification of the features to be controlled via flask. These are accompanied by Results in section VIII and lastly, conclusion and future plan can be seen in the section IX.

II. System Description

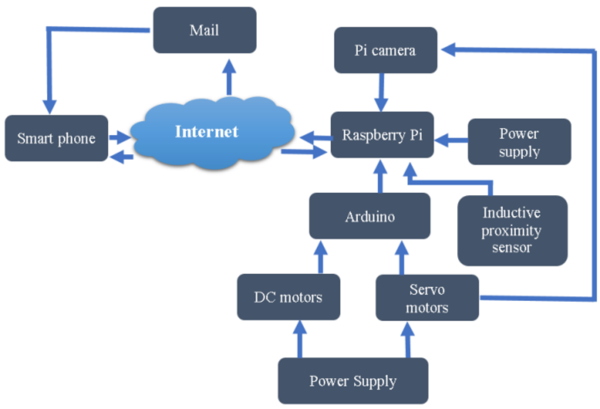

Fig 1. Block diagram of Smart Buggy

A. Object Detection

Object detection is one of the principal sections of Intelligent Rover. There are plenty of existing and ongoing researches to provide superior machine learning algorithms. The concept of neural network has been raised to mimic the human brain and assist in resolving issues associated with the areas of artificial intelligence, machine learning and deep learning. Generally, there are three neural networks, namely, Recurrent Neural Network (RNN), Deep Neural Networks (DNNs) and Convolutional Neural Network (CNN).

RNN saves the output of a specific layer and feeds this back to the input in order to predict the output of the layer. One of the major drawbacks of this network is the complication in training these networks. There are two classes of RNN, viz. Long Short-Term Memory Networks (LSTM) and Convolutional-LSTM (ConvLSTM). Both these classes have complex cells. [4] shows a Two-Path Convolutional LSTM Pyramid that stands out when compared to other RNNs. This network when blended with the CNN network offers object detection with great accuracy.

DNNs are Feed Forward Networks (FFNNs) where data gushes from the input layer to the output layer without rearward movement and the links between the layers are one way, that is, forward direction and they never connect a node again. DNNs have many downsides emphasizing on its low learning speed, need of huge data for training and the high costs associated with it.

CNNs are a class of DNNs and consists of consists of an input layer, multiple hidden layers, and an output layer. Convolutional layers convolve the input and forward the result to the succeeding layer. CNNs are categorized into Fully Convolutional Network (FCN), U-Net, Region Based CNNs (R-CNN), Fast R-CNN, Faster R-CNN, and You Look Only Once (YOLO) for object detection. Each of these is faster than other neural networks and their speed increases in the order as listed. The fastest of the CNNs is the yolo and is best suited to be run in raspberry pi. It is more than thousand times faster than R-CNN and hundred times faster than Fast R-CNN. Yolo comes with versions 1, 2, 3 and 4, each being better than other on the basis of speed and accuracy. The commonly used version is 3. Therefore, we have used yolov3 to detect guns and persons.

Yolo includes 53 convolutional layers which are also known as Darknet-53. For detection task, original architecture is stalked with 53 more layers that give 106 layers of architecture for YoloV3. The detections process occurs at 3 layers, specifically at layers 82, 94 and 106. Yolo V3 incorporates components that include residual blocks, skip connections and upsampling. Each and every convolutional layer is accompanied by batch normalization layer and Leaky ReLu activation function. Here, pooling layers are absent, alternatively, to downsample the feature maps, auxiliary convolutional layers having stride 2 are employed. This is done to hold back the loss of low-level features that were rejected by the pooling layers. Therefore, capturing low level feature maps assisted in enhancing the potential for identifying small objects.

a) Input to the network: The input is a batch of images of shape having n as the number of images, followed by width, height and number of channels RGB. Widths and heights are set as 416 or any other digits that are divisible by 32 without any remainder, for example, 608, 832 and 1024. The values of widths and heights are also called input network size. Therefore, the input image is of the shape (n, 416, 416, 3).

b) Detection at three scales: This step indicates detection places at the network. Detection is carried out in the layers 82, 94 and 106.

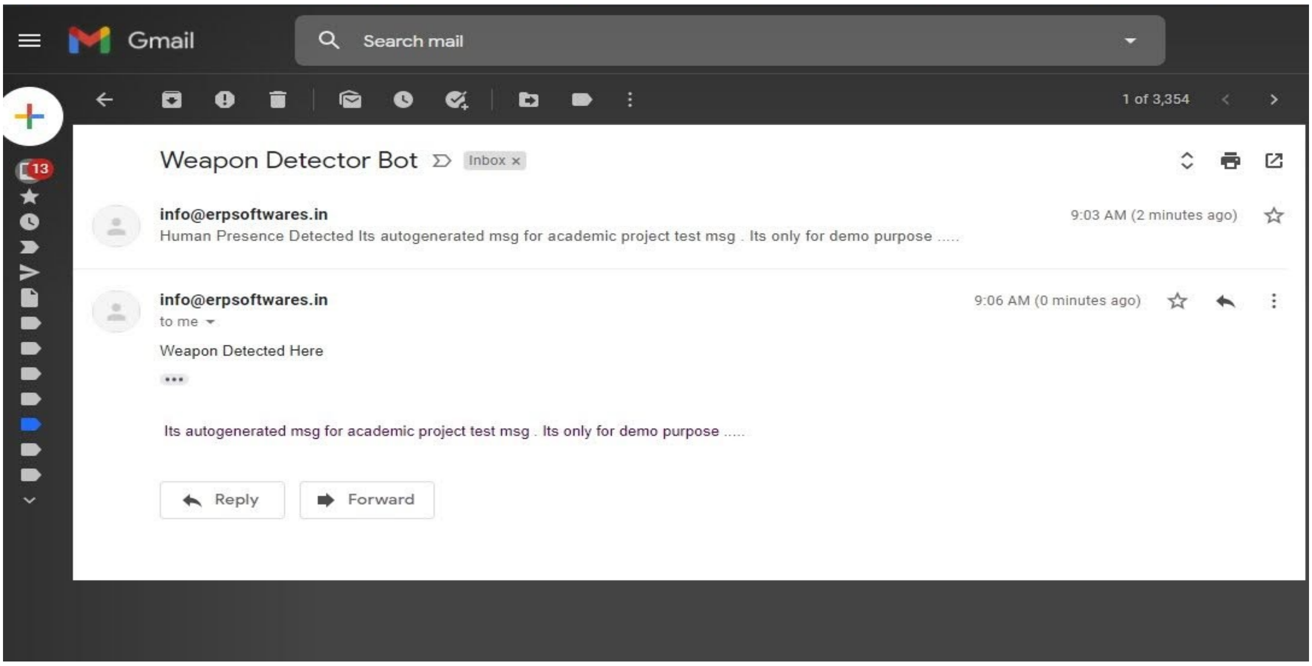

Fig.. 2. Email received that shows the presence of weapon and human.

The input image is downsampled by the network by factors of 32, 16 and 8 at these three layers of the network. These factors are also known as the stride of the network and they illustrate the smaller size of output at these three layers compared to the network input. For example, if 32 is the stride and input network size is (416 x 416), then the output will be of the size (13 x 13) which is responsible for detecting large objects. Consequently, for stride 16, output size is (26 x 26) and fore stride 8, output size is (52 x 52). The former is used to detect medium objects and the latter is used to detect small objects.

c) Detection Kernels: The output is obtained by applying 1 x 1 detection Kernels at these three distinct locations in the network. 1x1 convolutions are applied to downsample the input images (13 x 13), (26 x 26) and (52 x 52) which will produce feature maps with same spatial dimensions. The shape of the detection Kernels has its depths which are computed using the equation:

$$(B\times(5 + c)) \tag{1}$$

In (1), B represents the number of bounding boxes that every obtained feature maps can estimate. 3 bounding boxes are produced by YoloV3 for every cell corresponding to these feature maps. Therefore, value of B can be written as 3. Each bounding boxes has (5+c) attributes that describe the centre coordinates, widths and heights that are the dimensions of the bounding boxes, objectness score, and list of confidences for every class their bounding box belongs to.

d) Grid cells: Every cells of the feature map predicts an object via one of its bounding boxes when the centre of the object lies in the receptive field of that cell. Firstly, we have to conclude to which cell this bounding box belongs to. For this, we have divided the input image into a grid having dimensions similar to that of the feature map. In the fig. 3, the input image is of the dimension 416 x 416, with 32 as the network stride. Therefore, feature maps will have dimensions 13 x 13. Hence, input image is divided into 13 x 13 cells.

In the fig. 3, the cell responsible for detecting the object is marked in orange and contains the centre of ground truth box of the object which is marked in green. The orange cell falls in the ninth cell in the sixth row on the grid which is assigned for detecting the person. This cell is responsible for predicting 3 bounding boxes. We have used 5 anchor boxes which are defined by widths and heights. So the input image will be (m, 608, 608, 3) which is passed through deep CNN network which will lead to the encoding image of (m, 19, 19, 5, 85). The grid cell that detects the object will the one to which the centre of the object falls into. Every 19x19 cells encodes information of five boxes. We then find the product for each bounding box and then find the probability that the box contains the particular class. Since weapon and person are detected separately, we have only one class in each section with confidence of 0.5.

The dimensions of the bounding box are computed using log-space transformation that is applied to the output and further multiplying it with an anchor.

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

b_x and b_y are the x,y centre coordinates, b_w and b_h are the width and height of the prediction. The corresponding output of the network are represented by t_x ,t_y , t_w and t_h. The top-left coordinates are denoted by c_x and c_y while the dimensions of the anchor for the box are shown by p_w and p_h. Next, we have predicted centre coordinates via sigmoid function which resulted in the output values between 0 and 1.

e) Objectness score and confidences: Objectness score is the probability that an object is present inside a bounding box. It is approximately 1 for the orange and the grids in its neighbourhood, while it is approximately 0 for the corner grids. This score is also sent to the sigmoid function. Class confidences is the probabilities that the detected object belongs to a particular class.

f) Output processing: For the image of size 416 x 416, there are 1067 bounding boxes.

We have to reduce the detection down from 10647 to 1 because only one object is present in the image. For this, based on objectness score, we have filtered boxes, that is, boxes with scores under a threshold are ignored. Then, we have applied Non-maximum Suppression (NMS) which uses Intersection over Union(IoU) to eliminate the problems associated with multiple detection of the same image.

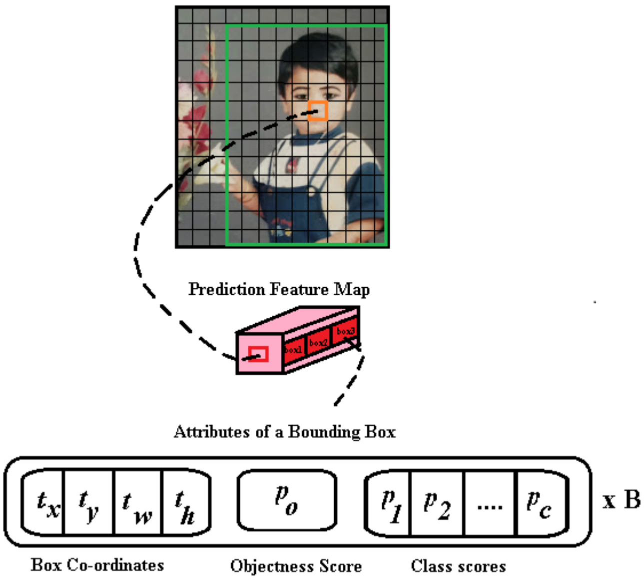

Fig. 3. Detection of person using YoloV3

The dataset was extracted from the Open imageV6 dataset. The dataset set for gun and person were collected separately. Around 800 images were collected for each object and were trained on Darknet-53 weights individually. Fig.4 shows weapon detection.

The detected objects will be saved and can be viewed in the webpage or in the saved folder. Also the user will be notified when a gun or a person is detected via E-mail as shown in fig. 2.

B. *Metal Detection*

The rover is capable of detecting metals with the range of 60mm using inductive proximity npn sensor. This section can be activated with live stream with or without object detection. Using this sensor, the metal detection process occurs without any contact with the metal as shown in fig. 5. Metals like iron, steel, copper and aluminium are detected using NPN inductive proximity sensor.

C. *Live Stream*

The Pi camera captures the real time video, deploys the yolov3 algorithm and live streams the scene as shown in fig 4. The Pi camera used for capturing the live video is OV5647 5MP day and night switching camera with IR LEDs to offer clear visuals during night time.

The Intelligent Rover comes with a separate section that focuses only on live stream that do not deploy the object detection algorithm. This section was predominantly provided to convert the Intelligent Rover into a surveillance system so that the applications can be spread into a wider range. With this section, the rover can transformed into a recue robot and can be employed in hard-to-reach areas or it can be used just for surveillance. The applications are not limited when it comes to security.

D. *Rover Movement*

Movement of the rover is one of the key sections of Intelligent Rover. We used two DC motors for the front wheels and two for the rear wheels. Each pair of DC motors is driven by L293-D H. The L293-D H and thus the DC motors are connected to Arduino Nano which in turn is connected to raspberry pi via the USB port. All the programs associated with the movement is implemented in the Arduino while the raspberry pi (model- 4B+) sends the commands according to the user needs. DC motors connected to the wheels sets the rover into motion with the ability to move forward, backwards, right and left.

E. *Swiveling the Camera*

Two servo motors are employed to swivel the camera horizontally and vertically. Like the DC motors, servo motors are connected to Arduino Nano. The servo motors employed for this mechanism have the ability to rotate from 0° to 180° according to the user. All the movements are under the control of the user. On each touch of tilt up control, the camera tilts 10 degree. Similarly, for down tilt, left pan and right pan, the camera does the corresponding motions with 10 degree difference. The PWM signals generated by the servo motors are responsible for the mechanical movement of the camera. The codes for swiveling the camera are programmed in Arduino and the mechanism is controlled through the commands sent from the raspberry pi.

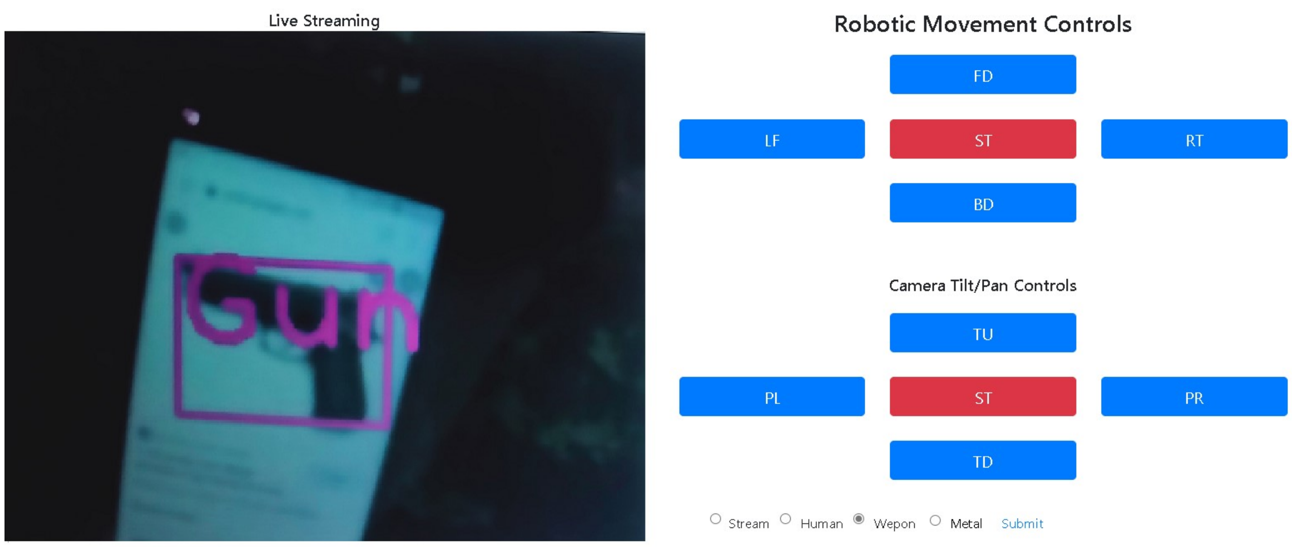

Fig. 4. Webpage to control the Intelligent Rover. The left side shows the detection of gun. The user can switch to person or detection or live stream only section.

F. Flask Micro Web

The Intelligent Rover is transformed into an IoT vehicle by enabling the controls through a webpage using flask micro web platform. Flask is a light WSGI web application framework that enables applications to be accessed and operated via a webpage.

Using flask micro web, a webpage is created that provides the user complete authority to control the Intelligent Rover. The commands that the user can provide includes the detection of guns, people or metal, switch to live stream section (without the object detection feature), swivel the camera horizontally and vertically and move the rover in desired directions. The webpage can be accessed via smartphone by entering the URL that includes the IP address of the connected network. Fig. 4 shows the webpage for the control.

III. IMPLEMENTATION

The heart of Intelligent Rover is Raspberry Pi 4B+ which comes with high RAM which helps in smooth detection of objects. Raspberry Pi was installed with Raspbian Buster. For object detection, OpenCV4 library was installed.

All the codes associated with the movement of the rover and camera were implemented on Arduino IDE and was uploaded into Arduino Nano.

The images used to train the model were taken from Open Images v6 dataset that offers a wide-range of images of different categories. The images were labelled and annotated by coding referred from GitHub.

Finally, the rover was converted to a wireless robotic device by replacing all the wired external power supplies as batteries. The Raspberry Pi was powered using 3A, 5V power bank and the servo and DC motors were powered using 8 AA batteries.

Fig.. 5. Metal detection

IV. RESULTS

The model of Intelligent Rover is shown in Fig. 6. The objective of Intelligent Rover is to aid the defense personnel to detect the intruders and weapons through a surveillance system that is of the size of a remote control car, thereby safeguarding their lives and protecting the country. This device is operated through a smartphone and by port forwarding the network, Intelligent Rover can be controlled from any spot of the world. The main features included in Intelligent Rover are listed below.

- Detection of guns and people using YoloV3 object detection algorithm.
- Streaming the captured video after deploying detection algorithm using the flask web app.
- Streaming real time video without deploying YoloV3 algorithm.
- Metal detection using proximity sensor
- Email alert when objects are detected
- Rover movements
- Camera movements
- Integrating all the features in Flask platform to be controlled through the webpage and controlled via smartphone.

Although YoloV3 algorithm runs faster than any other algorithm in raspberry pi, the detection process is slow during streaming. Yet it detects objects with greater accuracy. The device can also be used as a simple real-time video capturing device which can be used to surveillance hard-to-reach locations.

V. CONCLUSION

Using the thunderous computing network –the IOT, Intelligent Rover can be used to help the defense or security personnel. Its live stream only features extend its applications to wide-range from the simple surveillance system to a rescue device emphasizing on sequestered locations. The device is mainly used in surveillance and by switching to person detection, it can be deployed in houses, buildings and industries to detect intruders thereby eliminating the necessity of physical presence of humans in surveillance process.

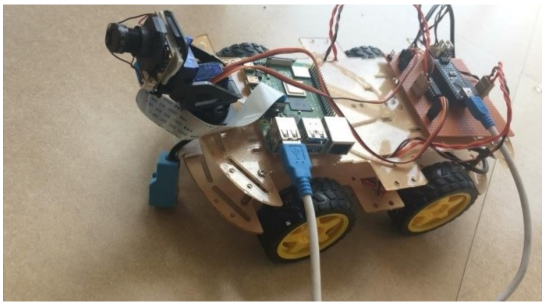

Fig.. 6. Model of Intelligent Rover

The absence of military based surveillance systems calls attention to Intelligent Rover. One of the main challenges faced during the development of the rover is on the selection of best algorithm for detection of guns and persons that can be run smoothly on raspberry pi. The training process was also difficult since it requires high RAM and GPU systems.

Intelligent Rover has a scope of wide extension. It can be used to detect more military based objects like tankers, grenades, missiles, etc. With better training platforms, all the objects can be trained together so that they can be detected in a single livestream. The Email feature can be upgraded to SMS so that the user can easily access the notifications. There is no boundary for the era of IoT. Anything can be done with a device employed in surveillance.

REFERENCES

[1] B. Bhomble, S. Katkar, D. Dhere, G. Arage, S. Bagwan and M. Jadhav, "IoT based smart sniper," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 557-561.

[2] R. P. and S. Bagwari, "IoT Based Military Assistance and Surveillance," 2018 International Conference on Intelligent Circuits and Systems (ICICS), 2018, pp. 340-344.I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3] C. Ke, "Military object detection using multiple information extracted from hyperspectral imagery," 2017 International Conference on Progress in Informatics and Computing (PIC), 2017, pp. 124-128.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] C. Zhang and J. Kim, "Video Object Detection With Two-Path Convolutional LSTM Pyramid," in IEEE Access, vol. 8, pp. 151681-151691, 2020.

[5] Harshitha R. and Muhammad Hameem Safwat Hussain, “Surveillance Robot Using Raspberry Pi and IoT”, IEEE Int. Conf on Design Innovations for 3Cs Compute Communicate Control (ICDI3C).

[6] K. H. S. Murugan, V. Jacintha and S. A. Shifani, "Security system using raspberry Pi," 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, 2017, pp. 863-864.

[7] N. Patil, S. Ambatkar and S. Kakde, "IoT based smart surveillance security system using raspberry Pi," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 0344-0348.

[8] V. P. Solanki and S. Deshmukh, "Wi-Fi Based Home Surveillance Bot," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-5.

[9] Marium F. Alvi, Faraz Junejo, Mashal W. Baloch and Ahmer J. Rajput, “RoboCop; A Robust Surveillance Robot”, IEEE Int. Conf on Advanced Research in Engineering Sciences (ARES).

[10] B. N. Rao, R. Sudheer, M. A. Sadhanala, V. Tibirisettti and S. Muggulla, "Movable Surveillance Camera using IoT and Raspberry Pi," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6.

[11] Ashish U. Bokade and V. R. Ratnaparkhe, "Video surveillance robot control using a smartphone and Raspberry pi", IEEE Int. Conf on Communication and Signal Processing (ICCSP).

[12] Lixianli, P. Wei, A. Jianyong and W. Ping, "The Application Research on Military Internet of Things," 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020, pp. 187-191.

[13] V. Gotarane and S. Raskar, "IoT Practices in Military Applications," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 891-894.