*IoT BASED ROVER FOR EXPLORATION*

*Abstract*— Objective: This project focuses on building a rover capable for rescue and exploration missions, where there is a risk to human life because of unknown dangers like toxic fumes, sudden avalanches, deep pits etc.

Findings: A tiny part of the world is always remained untouched since man had begun his exploration due to many unknown dangers. But now, robotic exploration is allowing us to virtually venture places that we never considered going. Be it archeological, metallurgical, or any rescue operations, unknown dangers to human life is always there. By using these kinds of rovers, the objective to explore unmanned places is achieved without posing any kind of threat to human life. Robots can consistently help human operators in dangerous tasks during rescue operations in several ways. Indeed, one of the main services that mobile robots can provide to rescue operators is to work as remote sensing devices reporting information from dangerous places that human operators cannot easily and/or safely reach.

Approach: The project has been implemented with several sensors to give a rough estimate about the area that is to be explored. These include temperature, humidity, toxic gases, a camera to give live feed of the location, a speaker to get noticed by the people in danger. The rover is controlled using a Raspberry Pi board, the controlling commands will be given via a html page. This link between the html page and Raspberry pi is created using a software called flask. The commands programmed will be used to activate and deactivate respective GPIO pins of the Pi which are responsible for the movement of the rover. A software called motion is used to access the USB webcam connected to the Pi. In the image processing part, which is also the major part of the project is carried by using an open source software called OpenCV. In OpenCV, real-time image processing is done using SSD (Single Shot Multi box Detector). The MobileNet used is pre-trained using COCO dataset and can recognize 20 objects with a mean average precision of 72.1%. The python script takes the video frames from the over and sends it to the neural network, wherein the predictions are made and displayed in a window, highlighting the predicted object and the confidence level.

*Keyword*—Raspberry pi, IOT, MobileNet, OpenCV, Image Processing

# **Introduction**

* 1. *Background*

In certain situations, data shows that more people who came to rescue died than the actual victims, reports from Occupational Safety and Health Administration, National Institute for Occupational Safety and Health stated that majority of the deaths are caused due to lack of knowledge of the area that the rescue rangers are dealing with. These mainly include of restricted entry and exits, unground vaults, manholes, tanks and breathing harmful fumes like methane, CO. In 2001 in Alabama, 12 miners attempting to rescue an injured miner after a coal mine explosion were killed by a second explosion.

Giving an idea about the hazards present in the area can reduce the risk to human life, rovers like this can come in hand to give a rough estimate about the possible dangers present in that area.

* 1. *Literature Survey*

A similar type of project has been done by the faculty and students of ABESIT Ghaziabad by using Arduinolilypad which is embedded with multiple sensors, and gesture control along with accelerometer to control the rover1. Also, RF was used for transmission and reception of the signals along with motor driver L293D was used to drive the motors. They proposed a low cost, low data rate design since they are using RF modules the communication range is limited to about 40 meters with good antenna design, and the execution of the signals is sometimes misinterpreted by the decoder due to addition of noise during the transmission.

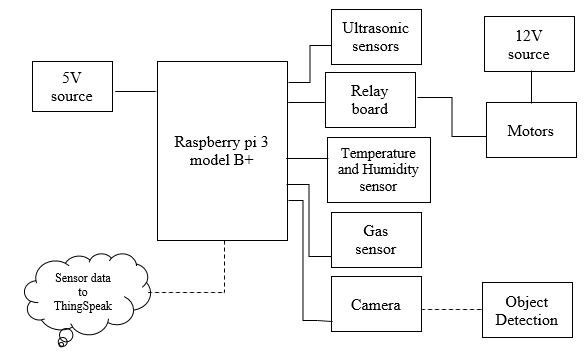
# **OBJECTIVE**

The following are the objectives of the rover:

* To build a prototype rover that has the capability of video and audio live stream the surroundings. //Ability to continuously livestream (video and audio) the surroundings.//
* Able to make the rover will be able to move on rough terrain. //Ability to move on rough terrain.//
* To collect data from sensors.//Data acquisition from onboard sensors.//
* Controlling the rover over a webpage or through a terminal.//To have the flexibility to control the rover either through terminal or a webpage//
* \*\*To show various parameters of the location in real time.\*\*

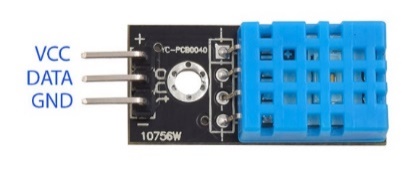
# **METHODOLOGY**

The rover has been designed with ultrasonic sensors, temperature and humidity sensor, CO gas detection sensor, and a camera mounted on a 180 degree servo motor for live video feed. All these sensors are connected to the GPIO pins of the raspberry pi 3 model B+. The data acquired from these sensors are shown in ThingSpeak platform to give a rough estimate about the location and the operator can decide his further steps in controlling of the bot. This rover is not built to operate autonomously thinking of unknown obstacles and the reactions it has to perform, which increases the complexity in coding and execution. This rover can be controlled by either using the terminal or through the webpage. OpenCV is used to detect human presence through the camera. Speaker is also connected to the audio jack of Pi to give a signal in the location mentioning the rover’s presence.

Coming to the motors used in the rover, these are 12V rated motors with almost 10Kg torque making it easy for the rigid and the heavy rover cross the rough terrain at ease. The power source for the rover is a 12V battery. The motors are controlled using a relay board \*\*unlike\*\* //instead of// motor driver ICs, as the maximum current output from the motor driver L298N is only 2Amps which is not sufficient to power the motors to their full potential which have the capacity to use current up to 4.5A

**Figure 3.1 Block diagram**

## Temperature and Humidity sensor (DHT11)



**Figure 3.2 DHT11 Sensor**

* Supply Voltage: +5 V
* Temperature range :0-50 °C error of ± 2 °C
* Humidity :20-90% RH ± 5% RH error

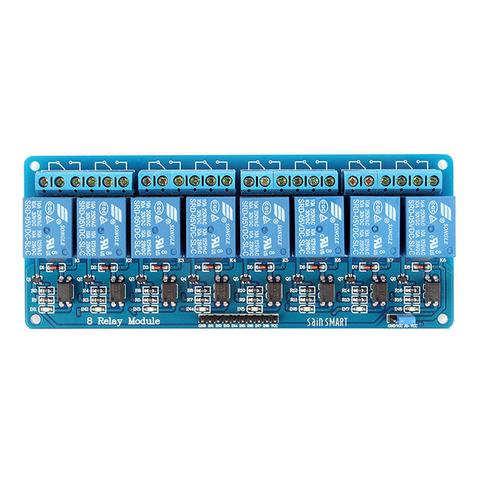
## Gas Sensor(MQ2)



**Figure 3.3 MQ-2 Sensor**

* Operating Voltage is +5V
* Can be used to Measure or detect LPG, Alcohol, Propane, Hydrogen, CO and even methane
* Analog output voltage: 0V to 5V
* Digital Output Voltage: 0V or 5V (TTL Logic)
* Preheat duration 20 seconds

## 8 Channel Relay board

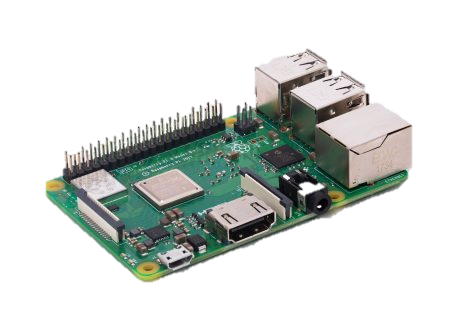


**Figure 3.4 8-Channel Relay board**

* Control Voltage: 5V DC.
* Max Control Capacity:10A at 250VAC or 10A at 30VDC.
* Low level triggered relay.
* Opto-Isolator.

## Raspberry Pi 3 B+

Raspberry Pi 3, Is a third generation Pi which is a best tool for students, embedded system design engineers and research scholars.



**Figure 3.5 Raspberry Pi 3 model B+**

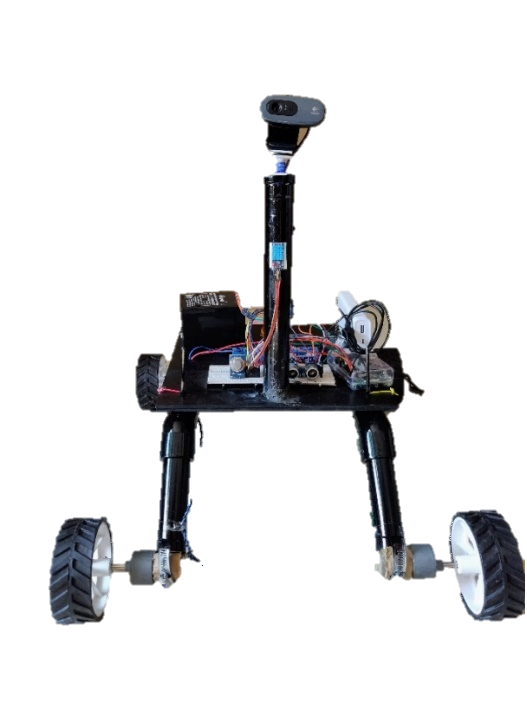
* Processor: 64-bit 1.4GHz ARM Cortex
* Power source: 5.1V-2.5A
* RAM: 1GB
* USB Ports: 4
* Storage: microSD

# **IMPLEMENTATION**

## This project can be broadly classified into two parts. One being the hardware part which includes the chasis and the structural part of the rover, other part consists everything about the software, this includes control of the rover through a webpage and image processing through OpenCV.

*4.1. Hardware*

The rover \*\*is\*\* //was// built on a custom design chassis, made of UPVC pipes and a wooden plank, to reduce the weight as much as possible but not sacrificing the strength and rigidity. It is a 4-wheel rocker design rover with the legs inclined at 45-degree angle to prevent toppling in case of rough terrain and also to efficiently use the torque of the powerful 12v DC motors.



**Figure 4.1 Hardware design of the Rover**

*4.2. Software*

In developing this rover, //standard terminal of raspberry pi and// various software products like OpenCV, motion and flask are used \*\*for image processing\*\*. Python coding was used to control the movement of the rover. The 8-channel relay board connected to the GPIO pins of the Raspberry Pi board are activated according to the required movement of the rover. A dedicated webpage was also developed for the rover control //and live feed//, this webpage has the commands to move the rover, a window to see what the camera on the rover shows and it also includes the commands to control the camera as well.

*4.3 Interfacing a USB Camera with Pi*

Motion is a software where the user can take video input from several inputs and configure them as needed. The user can change the resolution, fps, dimensions and the port to cast the feed. It is highly configurable software which take input from various types of cameras like Pi camera, webcam, network cameras. Using this software, the user can livestream camera feed directly via local IP-address of raspberry pi.

*4.4* *Building a Python Web Server with Flask*

To maneuver the rover, it would be better to design a combined environment of both vision and controls rather than running independent programs. //Also, running a standard program for rover control by multiple users would bottleneck the raspberry pi since every user has to login to the pi.// So, a web server \*\*has\*\* //had// been designed using a software named “Flask”, which has the capability to combine python programs with HTML and CSS and display on a webpage. The port created in the python program is accessible by any device connected to the same network. So, the rover can be controlled \*\*wirelessly\*\* //by any device with a web browser// as long as the controller and the rover are on the same network.

*4.5 Creating a webpage*

The user can add a new webpage to the server by creating a new route. In a web application, a route is a certain path into the website, determined by the URL the user types into their web browser’s address bar. It is up to the user which routes are enabled and what each of them does.

This route is made up of three parts:

1. @app.route(‘/’): This determines the entry point; the / means the root of the website, so http://192.168.43.236:5010.

2. def\_index (): This is the name given to the route and is called index, because it’s the index (or home page) of the website.

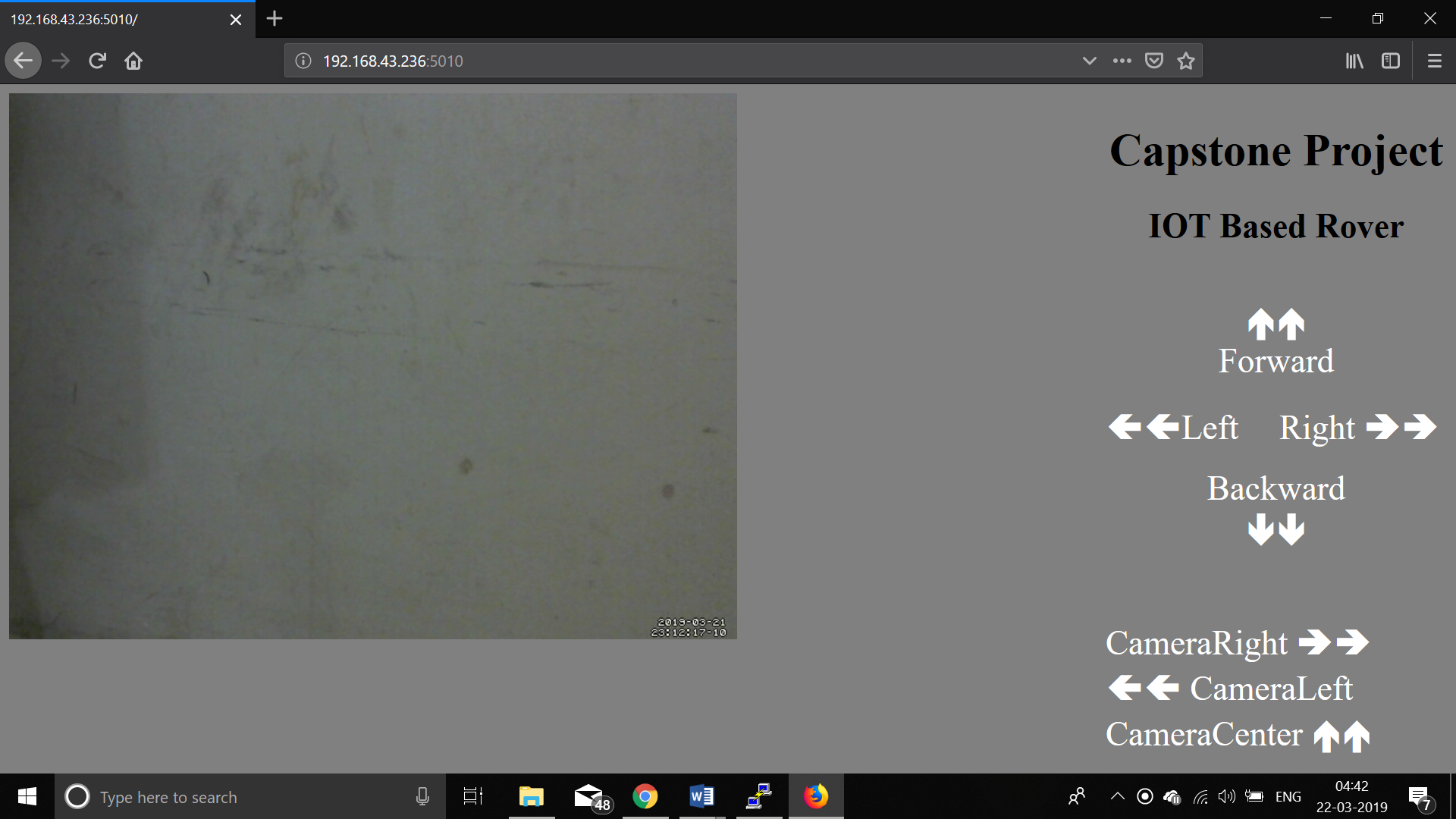
3. return ‘’: This is the content of the web page, which is returned when the user goes to this URL.

\*\*Now\*\* //For the rover control//, a series of General Purpose Input Output (GPIO) pins have been defined to control the relay to which the motors are connected, and the servo motor to which the camera is connected.

**pinList = [2,11,4,17,27,22,10,9,23,24,20,21]**

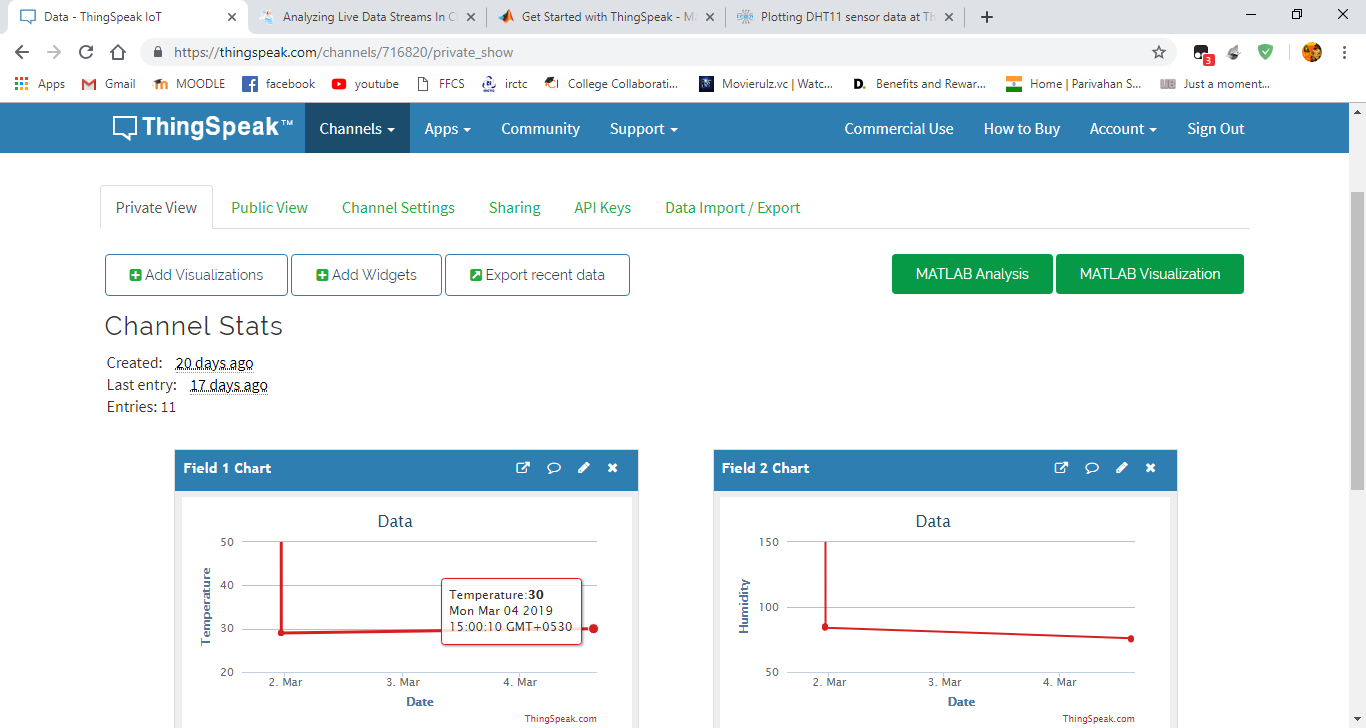
*4.6 HTML Webpage*

The webpage was designed in HTML and \*\*it is\*\* stored in a different directory named “templates”. \*\*For this, a directory named “templates” was created\*\*. With this, separate files with placeholders can be used for spots where the user wants to insert dynamic data. In order to compensate the lack of speed control, the script of the webpage \*\*is\*\* //was// created such that as long as the user presses and holds the button (hyperlink) on the webpage, the relays are accordingly controlled. Once the user retracts from the button, the relays are reset to HIGH. This also helps with instant stopping of the vehicle since the motors are forced to stop. In the HTML page that had been created, the livestream //was// taken from the “**URL: port**” (specified in the motion software “**http:192.168.43.236:8081**”) had been included and controls on the right side.



**Figure 4.2 HTML Webpage with controls**

DHT11 which gives the temperature and humidity data, MQ2 gas sensor which gives the presence of CO gas //and ultrasonic sensor which allows distance measurement//, \*\*these sensors\*\* are also interfaced with Pi the sensors data is sent to ThingSpeak, an online platform which gives a graphical representation of the data sent to it.



**Figure 4.3 Sensor Data into cloud (ThingSpeak)**

*4.7 Object detection*

For Object detection OpenCV \*\*has\*\* //was// been chosen to do the job, because of the available libraries with pre-trained models which are easy to implement. The program uses a MobileNet architecture and Single Shot Detector(SSD) together for fast and efficient object detection. The model used in this project post is a Caffe version of the original TensorFlow implementation by Howard et al. and it was trained by chuanqi305. The MobileNet SSD was trained on the COCO dataset (Common Objects in Context). It can detect 20 objects in images, including airplanes, bicycles, birds, boats, bottles, buses, cars, cats, chairs, cows, dining tables, dogs, horses, motorbikes, people, potted plants, sheep, sofas, trains, and monitors.

*Algorithm:*

* A class list and a color set is initialized.
* Each and every frame is looped over. First a frame from the stream is read, and resized. This frame is now converted into a blob using the Deep Neural Net(DNN) module.
* This blob is set as input to the neural network and this produces the detections.
* Next, the network looks at the confidence values and determines if it should draw a box and label the surrounding object or not.
* Now, the detections are looped over, keeping in mind that multiple objects can be detected in a single image. Also, it applies a check to the confidence (i.e., probability) associated with each detection.
* If the confidence is above minimum threshold, the class label index is extracted and the bounding box coordinates around the detected object are computed.
* The (x, y) coordinates of the box used for drawing the box and displaying text is extracted. Finally, the fps counter is updated.

# **C:\Users\Shivadeep\Downloads\Screenshot from 2019-04-02 09-39-35 (1).pngRESULTS AND DISCUSSIONS**

**Figure 5.1 Object detection**

The hardware and the software components of the rover work collectively to make it ideal for the rescue and exploration missions. The motors have enough torque to move over rough terrains. The sensor data also gets uploaded into ThingSpeak //either// on command //or at regular intervals//. The image processing part, which is a major part of the project is able to detect human presence. Raspberry pi is used to control the rover and camera movement, whereas the image processing part runs on laptop processor reducing the processing load over the Pi. Since the motors are controlled using a relay board unlike a motor driver IC(L298N), controlling the speed of the motors is not possible in this case. \*\*This is purposefully done as the motor driver IC limits the current to 2A which is not sufficient to drive the motors during heavy torque requirements.\*\*

It is successful in recognizing a person even when there is only a partial part of the person is seen. Which is the case in many rescue missions, generally people get stuck under the debris and the entire image of the person may not be available for recognition.

# **CONCLUSION AND FUTURE WORK**

# In this project, a rover to explore in areas with rough terrain and measure certain atmospheric parameters such as temperature, moisture and gas was demonstrated. The parameters were measured over vast areas (50m) owing to the long range of Wifi connectivity. This rover could be used for a wide variety of applications such as surveillance, home purposes, visual assistance in mines. The essential environmental parameters such as temperature, moisture and presence of gas were uploaded to ThingSpeak platform and analytics were displayed. Also, the basic movement controls of the rover were integrated into a website locally hosted by the RaspberryPi. The Rover is equipped with all the required sensors to help in rescue missions.

With the help of the rover, any visual or environmental information can be extracted and used for more IoT purposes. Any detection algorithm can be easily integrated and run on a host system by simply utilizing the vision from the rover. The scope of this project is to recognize people in the frames obtained from the live feed. Logging the obtained detections with time stamps would be a very useful future work. It would provide evidence for the observers involved in any kind of exploration. Also, implementing a GPS sensor with the ability to log coordinates of important locations (open areas or mines with high range connectivity) where any discovery is made and store the path it travelled in a given time interval would be very useful for data acquisition purposes.

# **REFERENCES**

[1] Internet of Robotic Things: concept Technologies, and Challenges, Partha pratim Ray.

[2] Implementation of Reconfigurable Robot Mechanism for Earthquake Rescue Operation, Swathiraj G.O.\* and Madhevan B.\*\* ***I J C T A****, 9(15), 2016, pp. 7411-7424 © International Science Press*

[3] A Cost Effective Way to Build a Web Controlled Search and CO Detector Rover, Ali Adib Arnab, Sheikh Sadia Afrin, F.M. Fahad and Hasan U. Zaman. *978-1-5090-4228-9/17/$31.00 ©2017 IEEE*

[4] Cost Effective Motion Based Stair Climbing Rover for Rescue Purpose, Sazid Al Ahsan1, Amir Hamza, Md. Hasibur Rahaman, Tasmiah Tamzid Anannya2, Md. Mahboob Karim *Proceedings of IEEE International Conference on Applied System Innovation 2018*

[5] Disaster Relief and Data Gathering Rover S.V.V. Srinivas, Aditya Kumar Singh, Aman Raj, Abhishek Shukla, Rachit Patel, Aviral Malay

[6] https://tutorials-raspberrypi.com/configure-and-read-out-the-raspberry-pi-gas-sensor-mq-x/

[7] https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor/raspberry-pi-thermal-camera