DEVELOPMENT OF AN EDGE AGENT FOR VIDEO ANALYTICS TO BE USED IN VIDEO MONITORING (VIMO) APPLICATION

[1]Prof. Abhilasha Kulkarni, [2]Nikita Modi, [3]Saloni Patil, [4]Sourabh Pawar, [5]Nikhil Thatte
[1]Assistant Professor, [2,3,4,5]Student
[1,2,3,4,5]Department of Computer Engineering,
[1,2,3,4,5]MMCOE, SPPU, Pune, India

Abstract : In the Development of an Edge Agent for Video Analytics to be used in Video Monitoring (VIMO) Application we have implemented two main modules. They are detection on a Region of Interest (ROI) and Integrated Detection. In Region of Interest, we have selected two regions and applied Person Detection on one region and Car Detection on the other region. Alerts are generated as soon as person or car is detected. Integrated detection includes integration of Face Detection, Object Detection and Motion Detection. When any detection occurs, a Output.avi file is generated which includes recording of real time detection results. Once the detection is completed, a JSON file containing the details of detection including timestamp, the event occurred, coordinates, the object detected is generated. These result files are uploaded on S3 (Simple Storage Service offered by Amazon AWS) using Message Queuing Telemetry Transport (MQTT) protocol. We can implement this application using a Webcam or a CCTV Camera connected to the same network as the host on which the application is running.

Keywords: *Video Analytics, Video Monitoring, Tkinter, ROI, MQTT*

I. INTRODUCTION

The application that we have developed can be mainly used for video surveillance. The main motivation behind this project is being able to survey a particular area and detect motion, facial features and objects in the environment. Surveillance can be carried out on the entire region covered by the camera or on a specified region of interest. The project has been implemented on the ubuntu platform using OpenCV and python script. The user interface is implemented using Tkinter.

In the first module we have implemented object detection in two Regions of Interest. This has been used for a CCTV feed obtained from a manufacturing plant. In the first ROI only, vehicles are allowed and if a person is detected an alert is generated. Similarly, in the second ROI only persons are allowed and if any vehicle is detected, a car alert is generated.

In the second module we have implemented an Integrated Detection which focuses up on Face Detection, Motion Detection and Object Detection. These detections are carried out simultaneously on the real time camera feed.

The outcome of this project will be that security systems will become more robust and effective in surveillance and identifying intruders.

II. METHODOLOGY

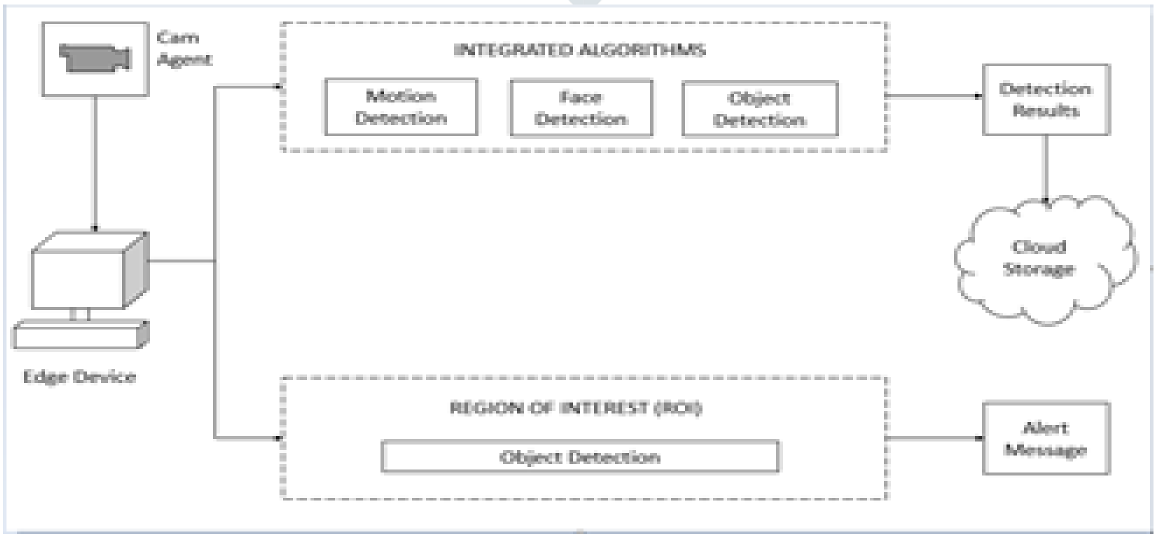


Figure 1. Architecture diagram of proposed system

This Application has been implemented with the help of OpenCV (version>=3.4.3). The programming language used is python and the complete application has been developed on Ubuntu 16.04. The user interface for the application is created using Tkinter.

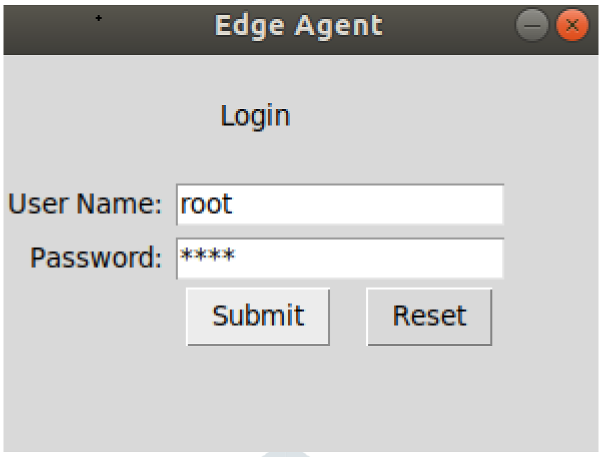


Figure 2. Login window created using Tkinter

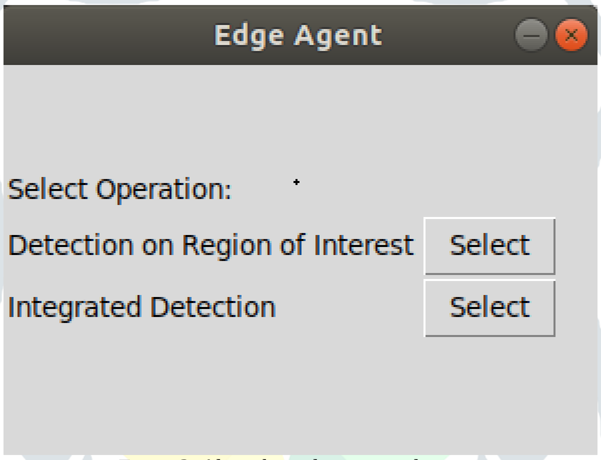


Figure 3. Algorithm selection window

Module 1: ROI

In this module we have two regions of interest in which we have implemented Object Detection. The input to this algorithm can be a live video stream from a webcam, a CCTV camera, or a static video. Object Detection has been implemented using a pretrained Caffe model called "MobileNet_SSD". We have limited the model to detect only Person and Vehicles. Object Detection for Vehicle has been implemented in the first Region. In this Region if a person is detected it displays a Person alert message on the window. Object Detection for Person has been implemented in the second Region. In this Region if a car is detected it displays a Car alert message on the window.

Module 2: Integrated Detection

This module is an integration of Face Detection, Object Detection and Motion Detection.

Face Detection

This algorithm has been developed using OpenCV's "haarcascade_frontalface_default.xml". In this algorithm whenever a front face is detected in a frame, a white bounding box is shown. The results are stored in a JSON file. The JSON file contains information about the event that has been detected, the timestamp and the X and Y coordinates. This result is uploaded onto S3 using MQTT protocol.

Algorithm:

```
Step 1: Start
Step 2: Import and load haarcascade_frontalface_default.xml
Step 3: for each frame captured:
check for images using detectMultiscale()
Step 4: if (face is found)
retrieve the coordinates
draw a rectangle showing the detections.
Step 5: Continue until key q is pressed.
Step 6: Create a json le that contains details of all the detections.
```

Step 7: Stop.

Object Detection

This algorithm is implemented using “MobileNet_SSD” which is a Caffe model. Here, we have limited the model to detect only Person. Whenever a person is detected in the frame, a random colored bounding box along with the label and confidence value is displayed. The results are stored in a JSON file. The JSON file contains information about the event that has been detected, the object detected and the timestamp. This result is uploaded onto S3 using MQTT protocol.

Algorithm:

```
Step 1: Start
Step 2: Load the pre-trained models and weights of CNN (Convolutional Neural Networks)
Step 3: Pass the video stream through the CNN and capture the detection results
Step 4: Compare the detection results with the threshold confidence.
if (detection results>threshold confidence)
Extract label of the object displaying the bounding box and the name of the object (label) as well the confidence level else
Discard the detection results
Step 5: Continue until key q is pressed.
Step 6: Stop
```

Motion Detection

This algorithm is based on the concept of reference frame. Here every new frame is compared with the previous frame and if there exists a difference motion is detected. A threshold has been set and if the motion detected is greater than the threshold, a green colored bounding box is shown. The results are stored in a JSON file. The JSON file contains information about the event that has been detected and the timestamp. This result is uploaded onto S3 using MQTT protocol.

Algorithm:

```
Step 1: Start
Step 2: Save a reference frame.
Step 3: Compare the frames from video stream to the reference frame.
if
(difference is found)
Display a bounding box tracking the motion
else
Continue analysis of the video frames
Step 4: Stop
```

All three of the above algorithms work simultaneously.

III. Results and Discussions

Module 1 Results:

On running the ROI module on a static video, the results observed are shown below.

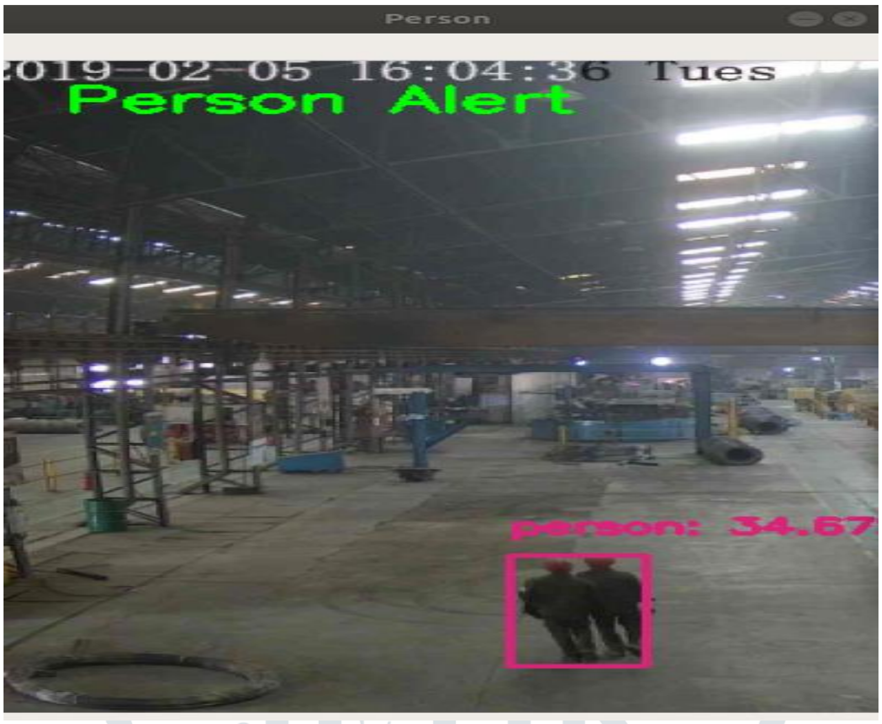


Figure 4. Results on ROI one

Module 2 Results:
On running the Integrated module on a webcam, the results obtained are shown below. The respective JSON files and their contents are also displayed. The FPS observed on running all three algorithms simultaneously is approximately 8.

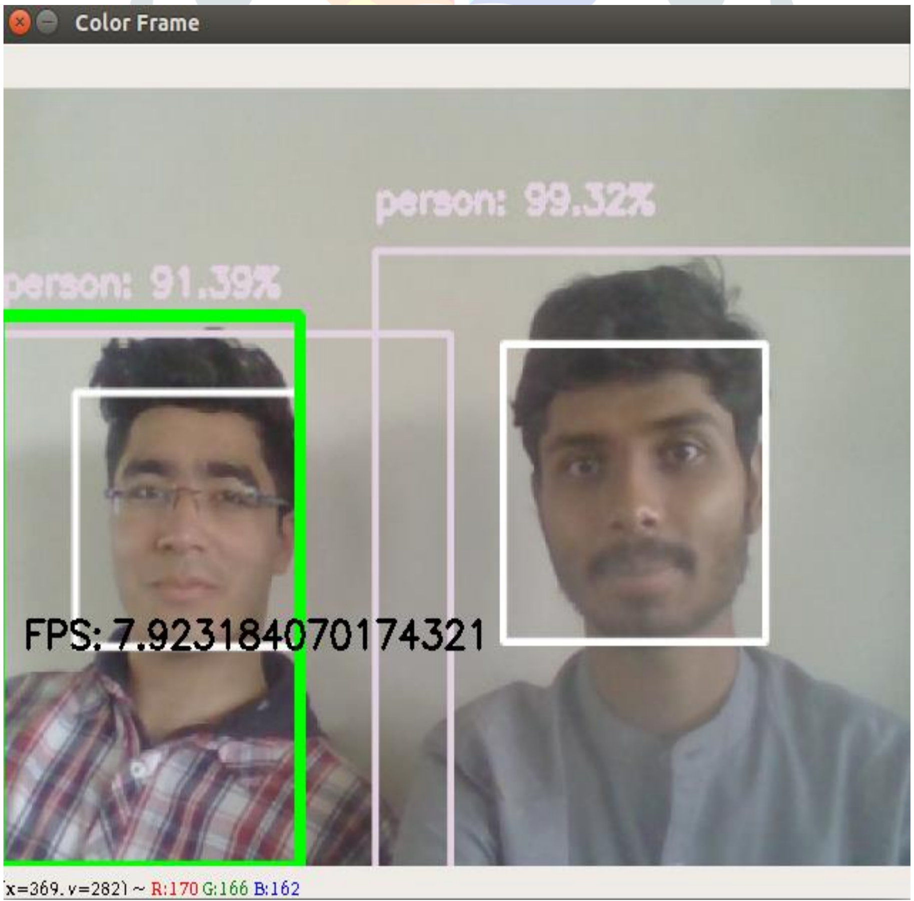
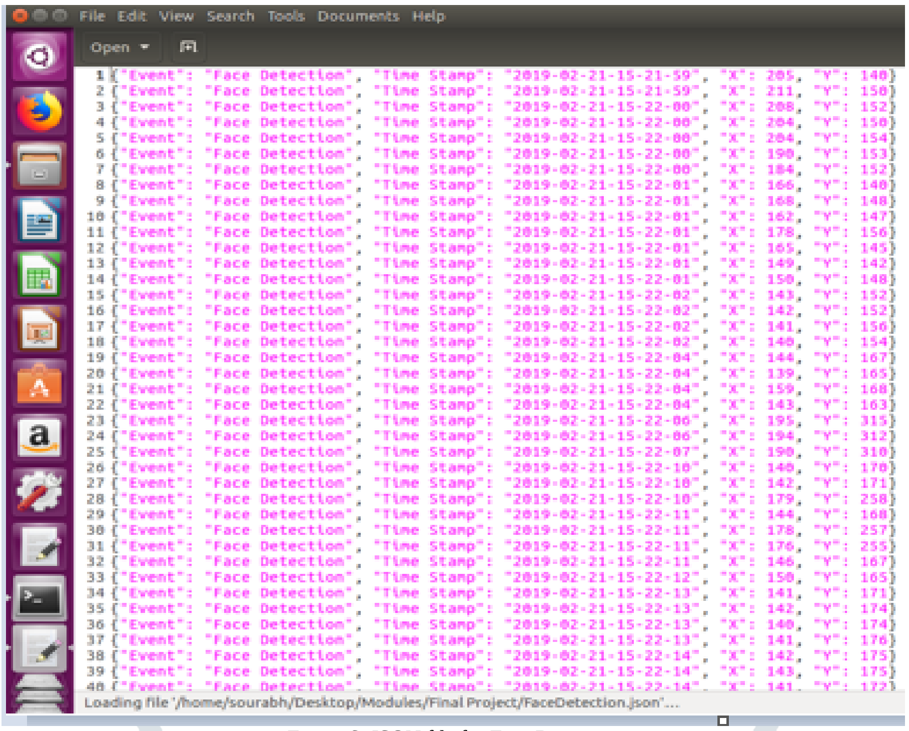


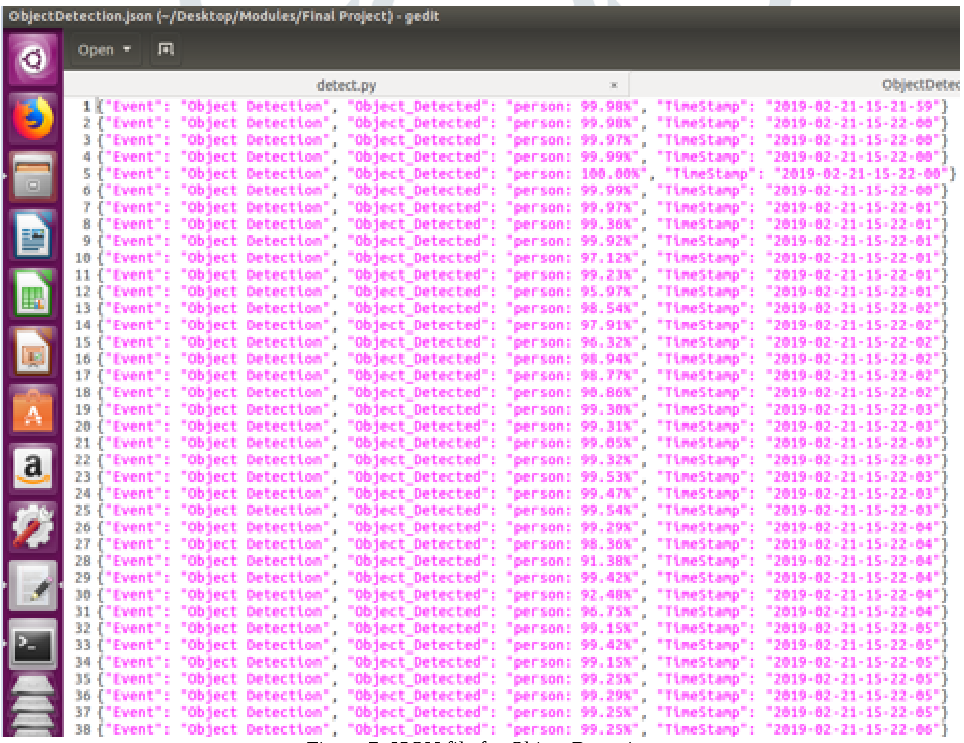
Figure 5. Outout for Integrated Detection



```
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-21-59", "X": 205, "Y": 140}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-21-59", "X": 211, "Y": 150}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-00", "X": 208, "Y": 152}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-00", "X": 204, "Y": 150}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-00", "X": 204, "Y": 154}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-00", "X": 190, "Y": 153}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-00", "X": 184, "Y": 152}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 166, "Y": 140}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 168, "Y": 148}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 162, "Y": 147}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 178, "Y": 156}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 165, "Y": 145}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 149, "Y": 142}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-01", "X": 150, "Y": 148}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-02", "X": 143, "Y": 152}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-02", "X": 142, "Y": 152}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-02", "X": 141, "Y": 156}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-02", "X": 140, "Y": 154}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-04", "X": 144, "Y": 167}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-04", "X": 139, "Y": 165}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-04", "X": 159, "Y": 168}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-04", "X": 143, "Y": 163}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-06", "X": 195, "Y": 315}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-06", "X": 194, "Y": 312}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-07", "X": 190, "Y": 310}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-10", "X": 140, "Y": 170}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-10", "X": 142, "Y": 171}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-10", "X": 179, "Y": 258}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-11", "X": 144, "Y": 168}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-11", "X": 178, "Y": 257}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-11", "X": 176, "Y": 255}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-11", "X": 146, "Y": 167}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-12", "X": 150, "Y": 165}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-13", "X": 141, "Y": 171}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-13", "X": 142, "Y": 174}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-13", "X": 140, "Y": 174}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-13", "X": 141, "Y": 176}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-14", "X": 142, "Y": 175}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-14", "X": 143, "Y": 175}
{"Event": "Face Detection", "Time Stamp": "2019-02-21-15-22-14", "X": 141, "Y": 172}
```

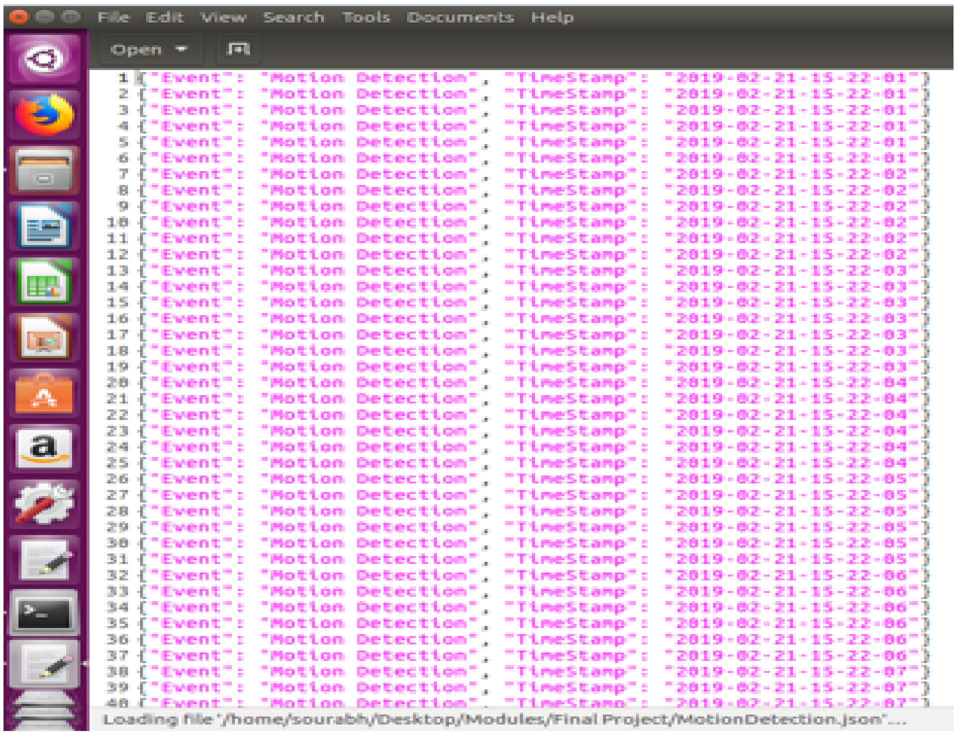
```
Loading file '/home/sourabh/Desktop/Modules/Final Project/FaceDetection.json'...
```

Figure 6. JSON file for Face Detection


ObjectDetection.json (~/Desktop/Modules/Final Project) - gedit
Open
detect.py
ObjectDetec

```
{"Event": "Object Detection", "Object_Detected": "person: 99.98%", "TimeStamp": "2019-02-21-15-21-59"}
{"Event": "Object Detection", "Object_Detected": "person: 99.98%", "TimeStamp": "2019-02-21-15-22-00"}
{"Event": "Object Detection", "Object_Detected": "person: 99.97%", "TimeStamp": "2019-02-21-15-22-00"}
{"Event": "Object Detection", "Object_Detected": "person: 99.99%", "TimeStamp": "2019-02-21-15-22-00"}
{"Event": "Object Detection", "Object_Detected": "person: 100.00%", "TimeStamp": "2019-02-21-15-22-00"}
{"Event": "Object Detection", "Object_Detected": "person: 99.99%", "TimeStamp": "2019-02-21-15-22-00"}
{"Event": "Object Detection", "Object_Detected": "person: 99.97%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 99.36%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 99.92%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 97.12%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 99.23%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 95.97%", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Object Detection", "Object_Detected": "person: 98.54%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 97.91%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 96.32%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 98.94%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 98.77%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 98.86%", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Object Detection", "Object_Detected": "person: 99.30%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.31%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.05%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.32%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.53%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.47%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.54%", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Object Detection", "Object_Detected": "person: 99.29%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 98.36%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 91.38%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 99.42%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 92.48%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 96.75%", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Object Detection", "Object_Detected": "person: 99.15%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.42%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.15%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.25%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.29%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.25%", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Object Detection", "Object_Detected": "person: 99.25%", "TimeStamp": "2019-02-21-15-22-06"}
```

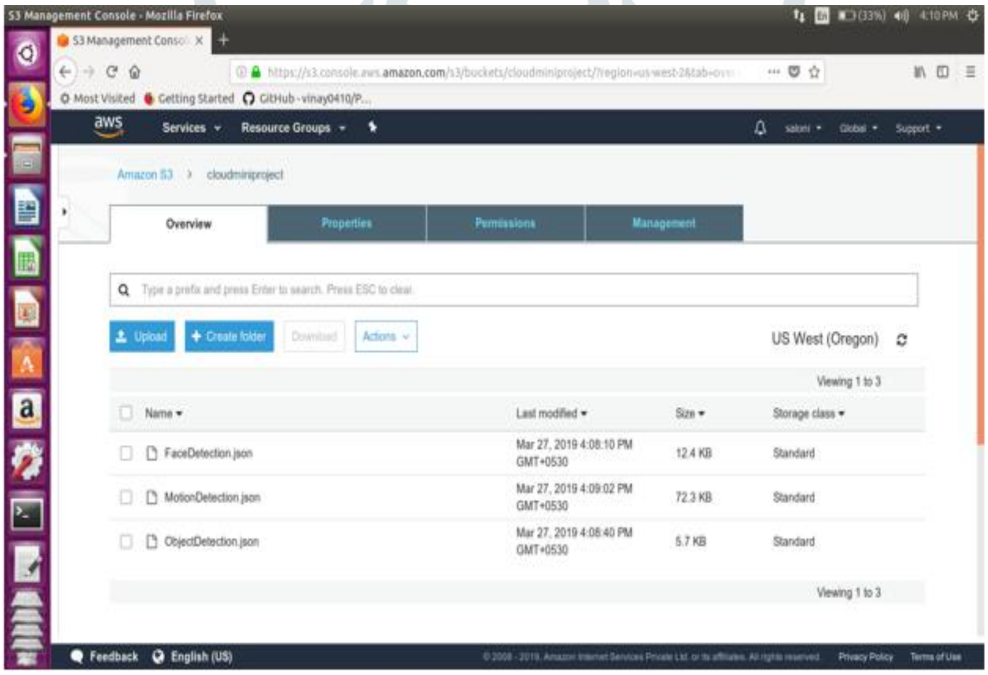
Figure 7. JSON file for Object Detection



```
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-01"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-02"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-03"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-04"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-05"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-06"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-07"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-07"}
{"Event": "Motion Detection", "TimeStamp": "2019-02-21-15-22-07"}
```

Loading file '/home/sourabh/Desktop/Modules/Final Project/MotionDetection.json'...

Figure 8. JSON file for Motion Detection



Name	Last modified	Size	Storage class
FaceDetection.json	Mar 27, 2019 4:08:10 PM GMT+0530	12.4 KB	Standard
MotionDetection.json	Mar 27, 2019 4:09:02 PM GMT+0530	72.3 KB	Standard
ObjectDetection.json	Mar 27, 2019 4:08:40 PM GMT+0530	5.7 KB	Standard

Figure 9. JSON files uploaded on S3

IV. Conclusion and Future Work

Using this application, we have fulfilled two requirements. We have provided surveillance for a specified region of interest and also provided integrated detection. We have uploaded the detection results on S3 which allows us to view them remotely.

Face detection can be improved to detect multiple faces from a side profile as well. Efforts can also be made to improve the fps obtained.

References

[1] S. V. Tathe, A. S. Narote, S. P. Narote , "Human Face Detection and Recognition in Videos" IEEE 2016.

[2] Shraddha G. Mhatre,Satishkumar Varma,Rupali Nikhare ,"Visual Surveillanceu sing Absolute Difference Motion Detection, IEEE 2015.
[3] Yanke Ma, Ti Peng,Tong Zhang, "A Fast and Robust face Detection and Tracking Algorithm" IEEE 2014.
[4] R.Raj Bharath, G.Dhivya , Moving Object Detection, Classication and its Parametric Evaluation ,IEEE 2014.