# Machine Learning & Data Sciensce Professional Projects

Amir Jamali

---

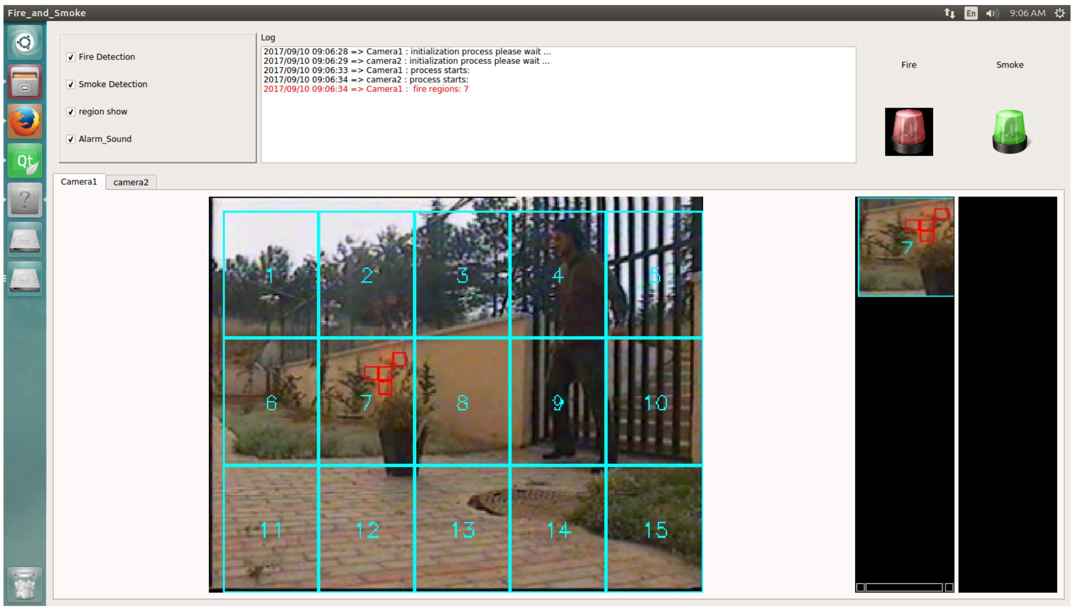
## introduction

I have been Researcher & Developer since 2014 during in which I've conducted some computer vision applications such as Fire & Smoke Detection, Intrusion Detection, Face Recognition, People Counting, abandoned object, Camera Tampering and have contributed in some project such as Car License Plate Recognition, Anomaly Detection(Network Cyber Security) with C++ & Python.
I have been trying to develop industrial software which makes me tackle challenges in the real condition to enrich my experience in machine learning. All of the following computer vision applications mentioned, receive an online stream from IP camera and processes any frame in real-time. In the following, I'll give a brief description of each application but not in detail because of the company's right.

---

## Fire & Smoke Detection

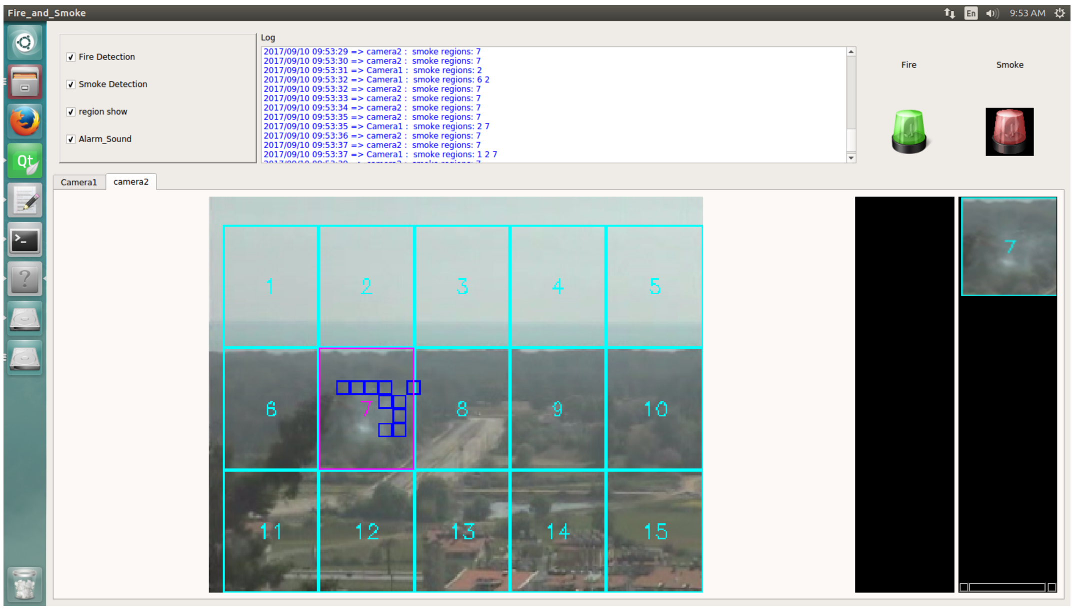
This application is developed in a multi-threaded way to process each camera stream independently. Each thread extracts some features from blocks with specific size and gives them to a classifier with some constraint. After then if the number of candidate block for fire and smoke becomes larger than a specific threshold it Alarms fire or smoke at the corresponding regions, and, shows them in the sidebar, sends SMS alarm using GSM module and reports camera number, event time, region number and type of alert to the remote database.



✓ Fire Detection
✓ Smoke Detection
✓ region show
✓ Alarm_Sound

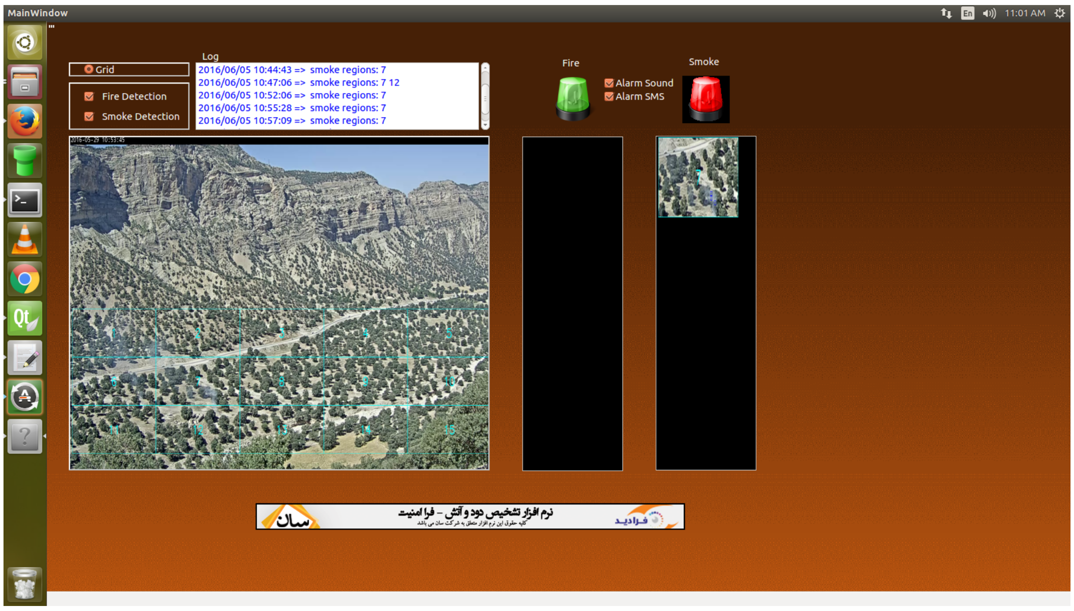
Log

```
2017/09/10 09:06:28 => Camera1 : initialization process please wait ...
2017/09/10 09:06:29 => camera2 : initialization process please wait ...
2017/09/10 09:06:33 => Camera1 : process starts:
2017/09/10 09:06:34 => camera2 : process starts:
2017/09/10 09:06:34 => Camera1 :  fire regions: 7
```

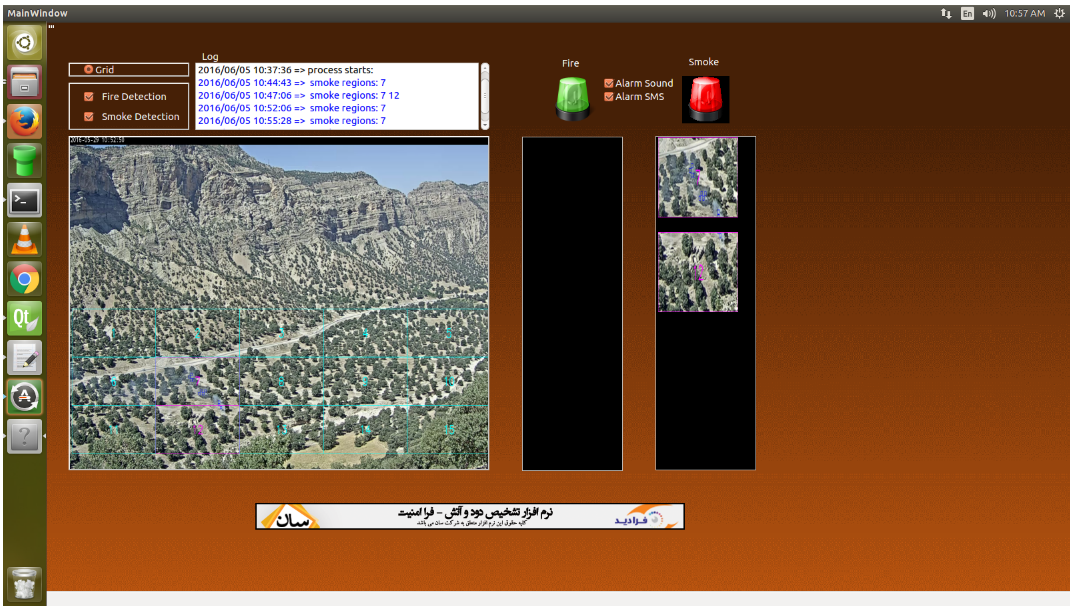


```
2017/09/10 09:53:29 => camera2 :  smoke regions: 7
2017/09/10 09:53:30 => camera2 :  smoke regions: 7
2017/09/10 09:53:31 => Camera1 :  smoke regions: 2
2017/09/10 09:53:32 => Camera1 :  smoke regions: 6 2
2017/09/10 09:53:32 => camera2 :  smoke regions: 7
2017/09/10 09:53:33 => camera2 :  smoke regions: 7
2017/09/10 09:53:34 => camera2 :  smoke regions: 7
2017/09/10 09:53:35 => camera2 :  smoke regions: 7
2017/09/10 09:53:35 => Camera1 :  smoke regions: 2 7
2017/09/10 09:53:36 => camera2 :  smoke regions: 7
2017/09/10 09:53:37 => camera2 :  smoke regions: 7
2017/09/10 09:53:37 => Camera1 :  smoke regions: 1 2 7
```

This application is successfully tested as fire & smoke detector in real condition at Department of Environment of Iran fire extinguisher maneuver in Ilam state. The Software had been running since 2 hours before maneuver started without any false alarm, and when smoke started, it began to send an alert in real-time immediately.



```
2016/06/05 10:44:43 =>  smoke regions: 7
2016/06/05 10:47:06 =>  smoke regions: 7 12
2016/06/05 10:52:06 =>  smoke regions: 7
2016/06/05 10:55:28 =>  smoke regions: 7
2016/06/05 10:57:09 =>  smoke regions: 7
```



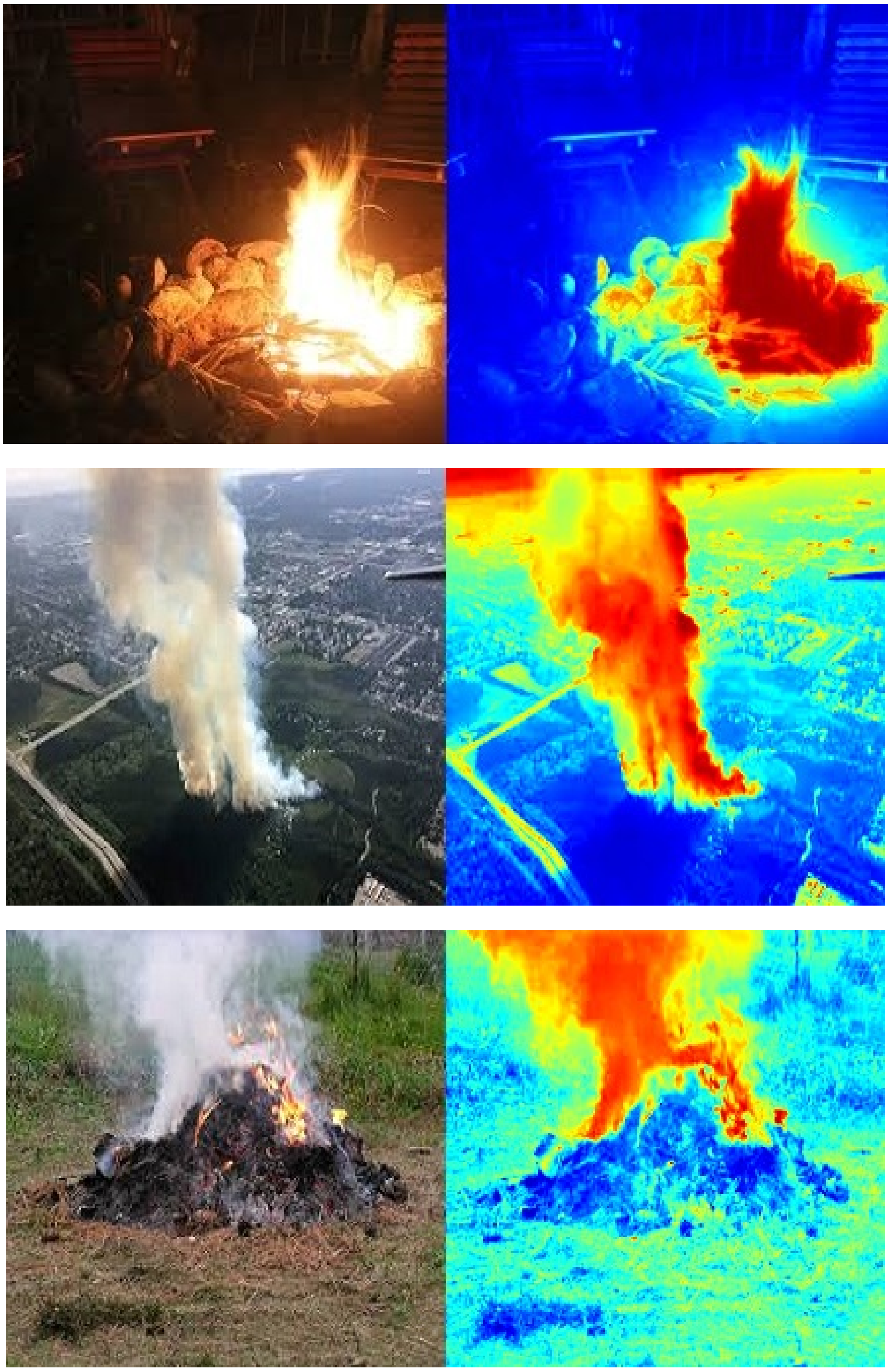
```
2016/06/05 10:37:36 => process starts:
2016/06/05 10:44:43 =>  smoke regions: 7
2016/06/05 10:47:06 =>  smoke regions: 7 12
2016/06/05 10:52:06 =>  smoke regions: 7
2016/06/05 10:55:28 =>  smoke regions: 7
```

To see a short clip related to maneuvering, please click Here

---

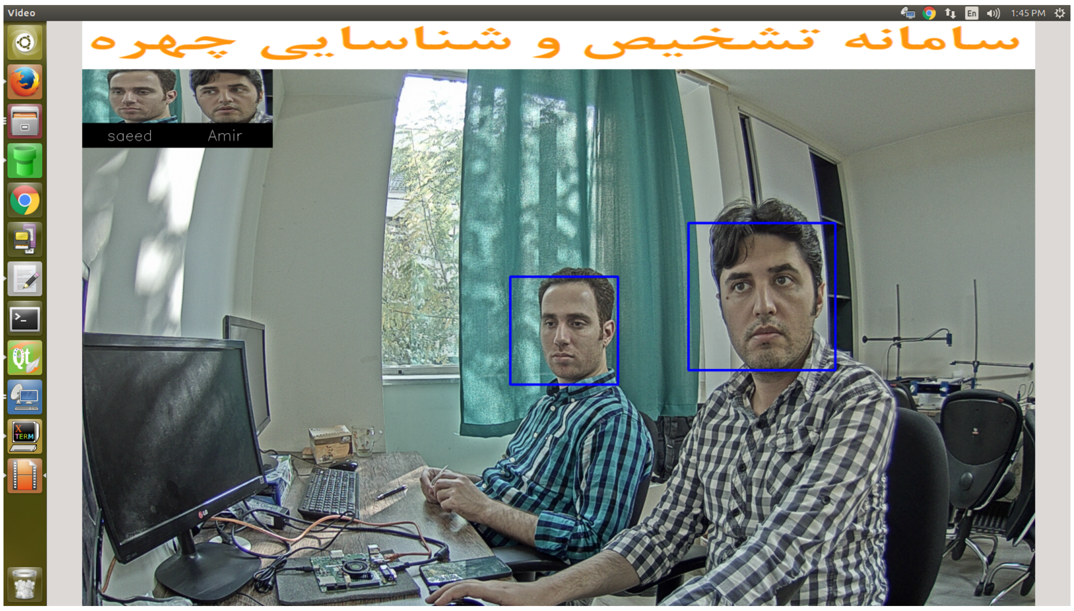
## Fire & Smoke Detection using Deep Convolutional Neural Network

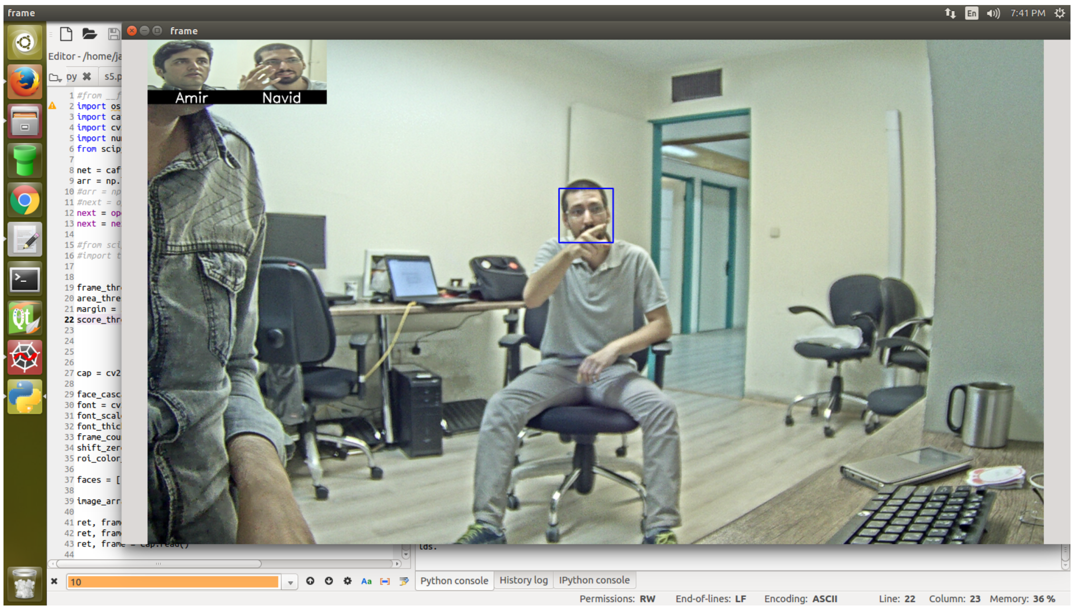
I gathered 12125 images which included fire, smoke, or both scenes as positive samples and 16631 photos as negative ones from internet. After then I froze all layers of Resnet50 save that last layer, and fine-tuned that layer for my data. finally I localized fire and smoke in image by generating heatmap which was synthesized by integrating feature map with respect to their corresponding linear weight means.



## Face Recognition & Face Grouping

we used a Resent34 fine-tuned by 5000 class(person) and 50000 samples and Mxnet-Face as feature extractors for face recognition and face grouping under thirty-degree faces in camera stream which yields outstanding results in real-time on GPU.





## Crowd Estimation

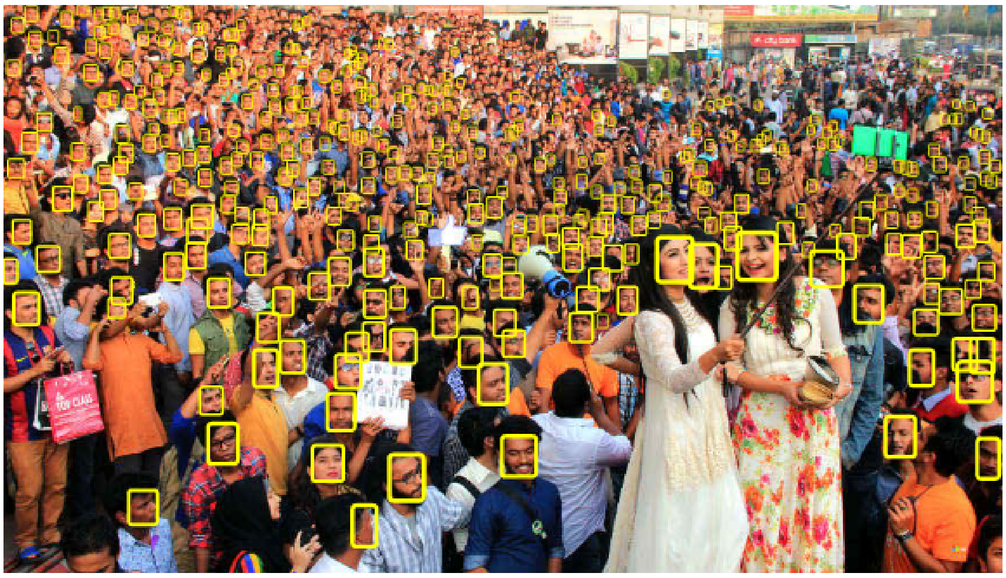
This application was developed in two approaches:

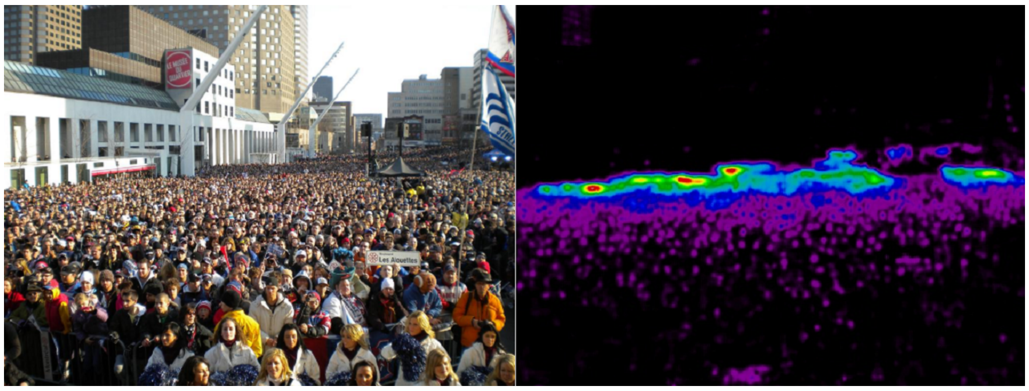
**Using the face detector represented by Mxnet-Face**

- it reports the number of faces detected in the image

**Using and fine-tuning the model proposed by [1]A. Sindagi**

- For the densely crowded case in which people's faces are not large enough to detect by the face detector this model generates dense heatmap estimation of the crowd and approximates the number of people as well.

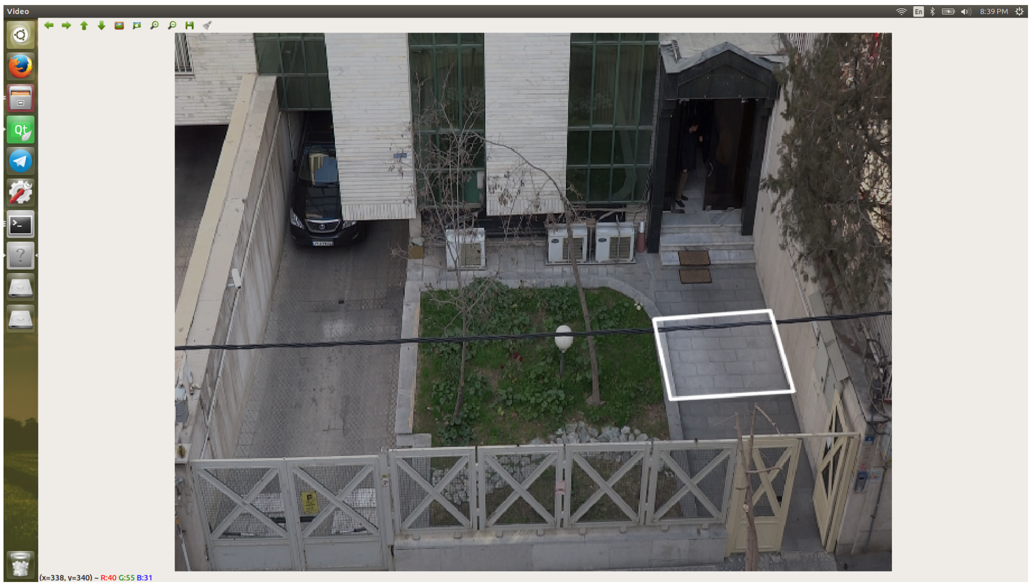


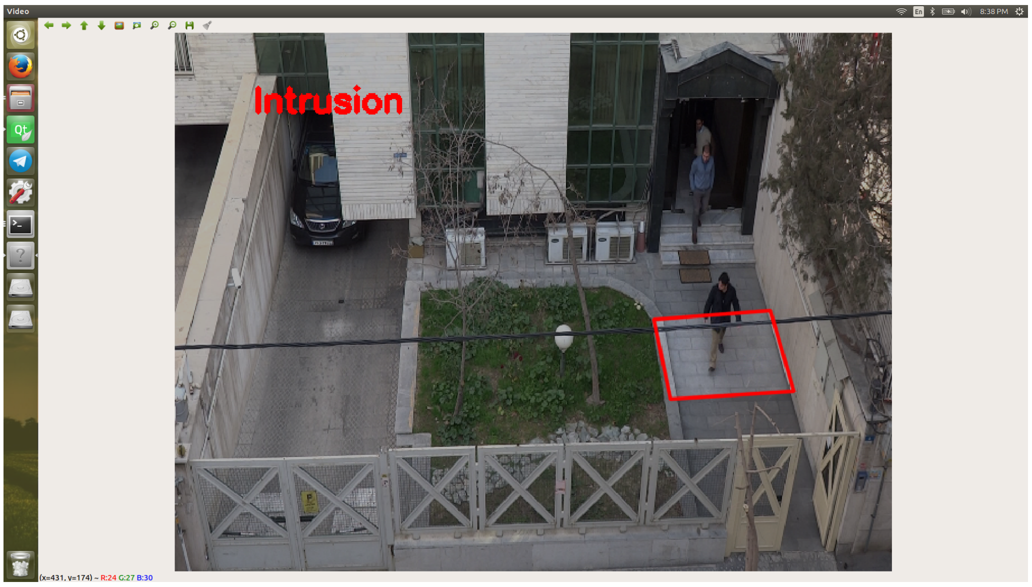


[1]CNN-based Cascaded Multi-task Learning of High-level Prior and Density Estimation for Crowd Counting

## Intrusion Detection

This application alerts any human entrance into a forbidden area([2]ROI) which is given by the user. Cascade people detection was used to reduce false alarms to detect solely human as a moving object.

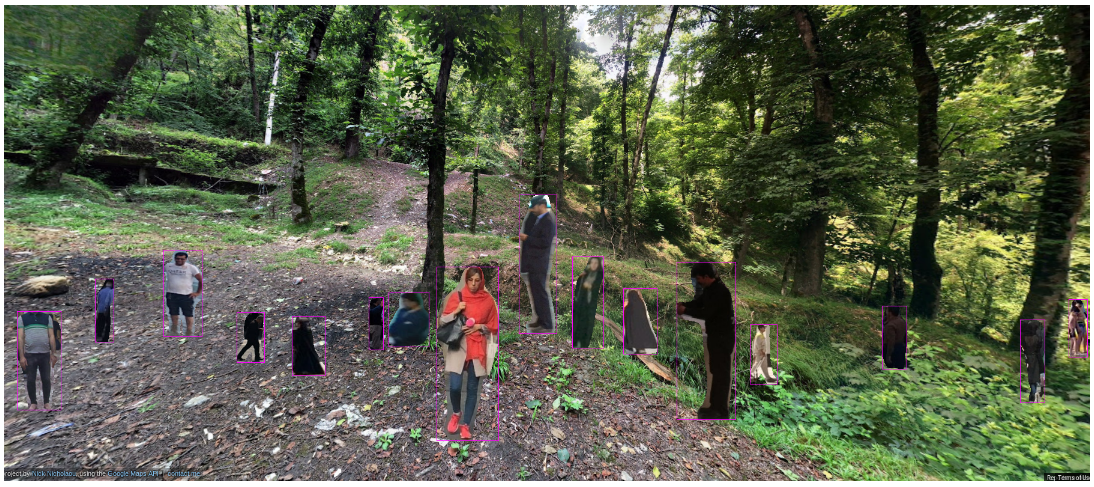




[2]Region Of Interest

## Pedestrian Detection

It goes without saying that pedestrian detection is a highly challenging problem in computer vision. Its performance is generally susceptible to people clothing. Since the type of clothing ,especially for women, are different in the middle east compared to the other part of the world. I decided to generate my own dataset for this region. First I gathered 7000 images of Google Street View of my country including pedestrian and 10000 images without pedestrian as well. Second, I used a pre-trained mask-rcnn-150 model of Tensorflow Object Detection API to extract mask of pedestrians in images; that is because this model is highly accurate even though is slow. After then I synthesize 70000 images with random background and various pedestrians. Finally I trained a faster-rcnn-50 pre-trained model with my data which yield a great result(fast and accurate)



## Car License Plate Recognition

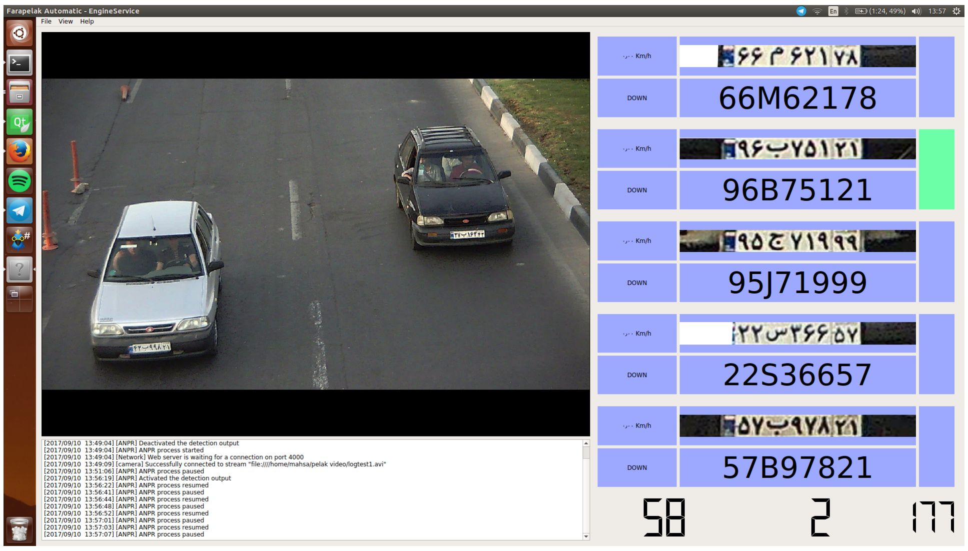
This application uses a Deep Convolutional Neural Network to recognize license plate. It is developed in two forms:

**Fixed IP Camera**

- For road, highways or parking entrance

**Mobile IP Camera**

- In this case, the camera is located on a traffic police car which moves through street and alleys in the city and application reports violator's car plate numbers, and it's GPS position(by serial GPS receiver) to a remote database.



```
[2017/09/10  13:49:04] [ANPR] Deactivated the detection output
[2017/09/10  13:49:04] [ANPR] ANPR process started
[2017/09/10  13:49:04] [Network] Web server is waiting for a connection on port 4000
[2017/09/10  13:49:09] [camera] Successfully connected to stream "file:////home/mahsa/pelak video/logtest1.avi"
[2017/09/10  13:51:06] [ANPR] ANPR process paused
[2017/09/10  13:56:19] [ANPR] Activated the detection output
[2017/09/10  13:56:22] [ANPR] ANPR process resumed
[2017/09/10  13:56:41] [ANPR] ANPR process paused
[2017/09/10  13:56:44] [ANPR] ANPR process resumed
[2017/09/10  13:56:48] [ANPR] ANPR process paused
[2017/09/10  13:56:52] [ANPR] ANPR process resumed
[2017/09/10  13:57:01] [ANPR] ANPR process paused
[2017/09/10  13:57:03] [ANPR] ANPR process resumed
[2017/09/10  13:57:07] [ANPR] ANPR process paused
```

---

## People Counting

In this application the camera is located at the top of entrance and application counts the number of people who come in or out. It is developed in two forms:

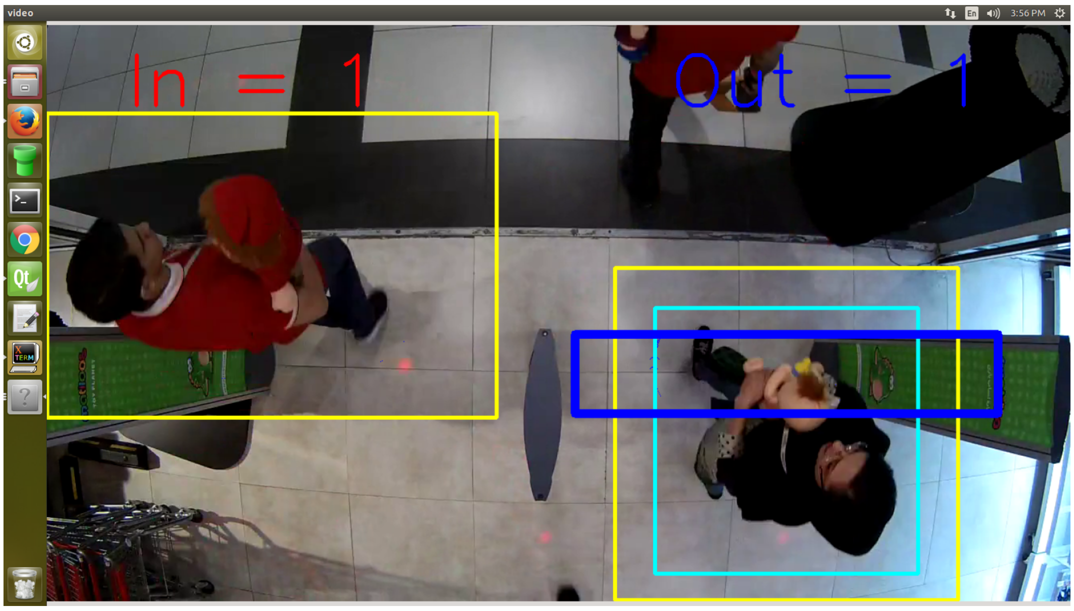
**RGB Camera**

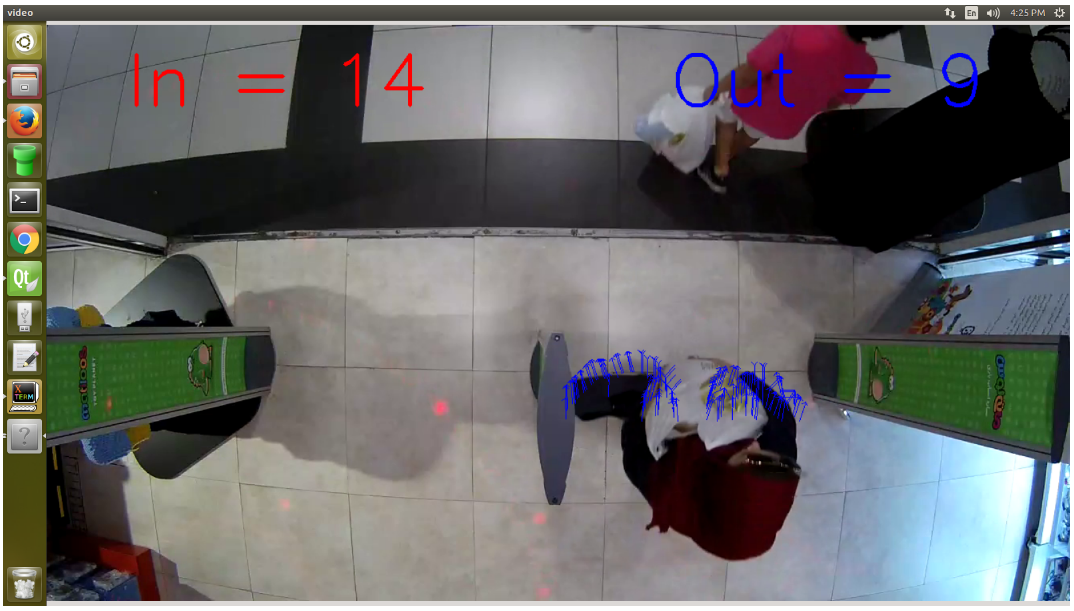
- which uses blob-base image processing techniques

https://www.overleaf.com/project/59dc6b3ba58aaa767a02d4a2

**Kinect Camera**

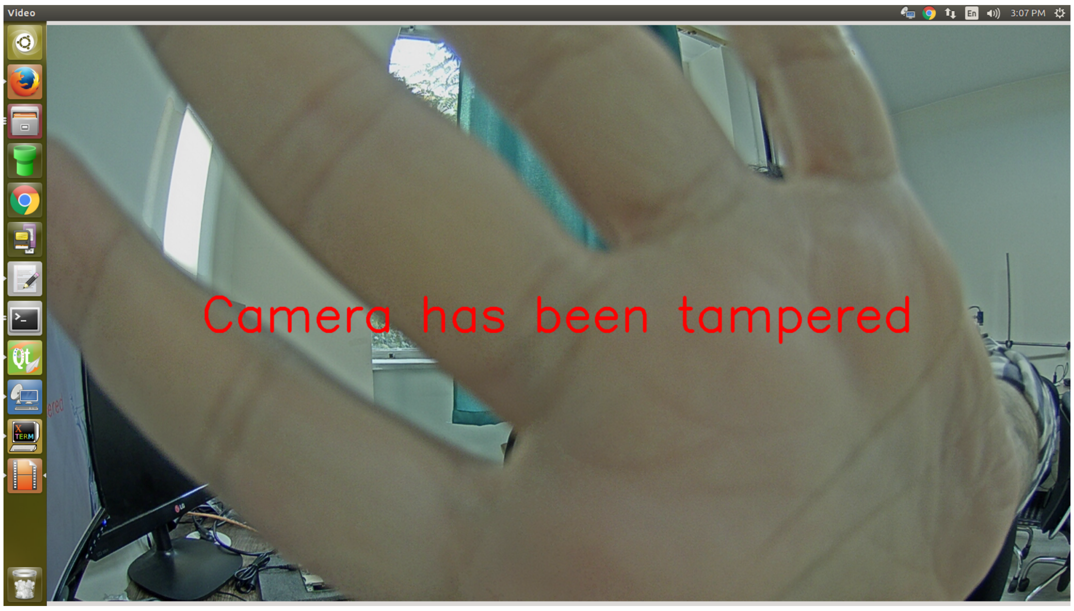
- which used infrared depth map image to extract people's head





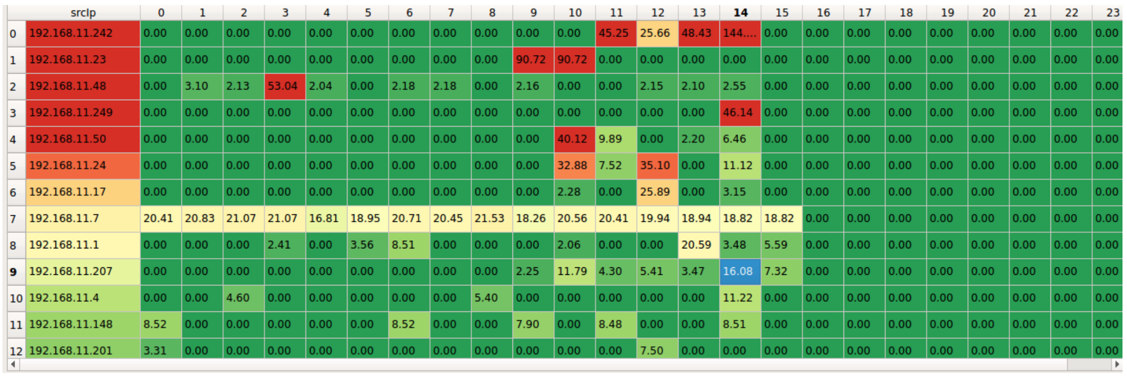
## Camera Tampering

This application is developed to protect surveillance cameras from tampering, and it alarms when someone moves or occludes camera.



## Anomaly Detection(Network Cyber Security)

This application presents a prototype for anomaly detection in network flow(ipfix data). It trains an auto encoder model with respect of the traffic behavior of each IP on each day of the week, so the traffic behaviors excluded from that are considered as anomaly behavior. The following image is an example report of this application. Higher scores(redder)specify more anomaly.

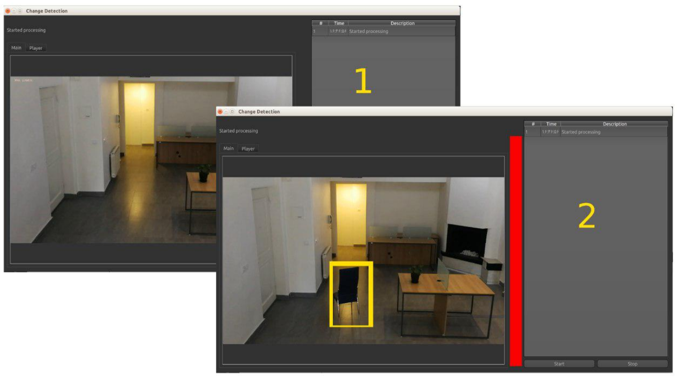


| | srcIp | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | **14** | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 192.168.11.242 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 45.25 | 25.66 | 48.43 | 144.... | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 192.168.11.23 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 90.72 | 90.72 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 192.168.11.48 | 0.00 | 3.10 | 2.13 | 53.04 | 2.04 | 0.00 | 2.18 | 2.18 | 0.00 | 2.16 | 0.00 | 0.00 | 2.15 | 2.10 | 2.55 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 192.168.11.249 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 46.14 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 192.168.11.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 40.12 | 9.89 | 0.00 | 2.20 | 6.46 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5 | 192.168.11.24 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 32.88 | 7.52 | 35.10 | 0.00 | 11.12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6 | 192.168.11.17 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 2.28 | 0.00 | 25.89 | 0.00 | 3.15 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7 | 192.168.11.7 | 20.41 | 20.83 | 21.07 | 21.07 | 16.81 | 18.95 | 20.71 | 20.45 | 21.53 | 18.26 | 20.56 | 20.41 | 19.94 | 18.94 | 18.82 | 18.82 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8 | 192.168.11.1 | 0.00 | 0.00 | 0.00 | 2.41 | 0.00 | 3.56 | 8.51 | 0.00 | 0.00 | 0.00 | 2.06 | 0.00 | 0.00 | 20.59 | 3.48 | 5.59 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **9** | 192.168.11.207 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 2.25 | 11.79 | 4.30 | 5.41 | 3.47 | 16.08 | 7.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10 | 192.168.11.4 | 0.00 | 0.00 | 4.60 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 11.22 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 11 | 192.168.11.148 | 8.52 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 8.52 | 0.00 | 0.00 | 7.90 | 0.00 | 8.48 | 0.00 | 0.00 | 8.51 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 12 | 192.168.11.201 | 3.31 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 7.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

---

## Abandon Object Detection

If someone leaves something behind in a particular area after a specific time this application alerts


Change Detection
Started processing
Main
Player
Time
Description
1
2
Start
Stop