Project Phase-II

Understanding Pedestrian Behaviour to Improve Working of Autonomous Vehicles

Pratibha Agarwal	1PE15IS076
Srikrishna CN	1PE15IS102
Malvika Diwan	1PE15IS125

May 8, 2019

Guide: Prof M S Anand

Problem definition

A major dilemma faced by autonomous cars is understanding the intention of other road users and communicating with them.

To investigate one aspect of this, specifically pedestrian crossing behavior, we shall analyze a large dataset of pedestrian samples at crosswalks under various conditions (e.g., weather, time of the day) and in different types of roads.

Using the data, we shall analyze pedestrian behavior from two different perspectives: the way they communicate with drivers prior to crossing and the factors that influence their behavior.

Problem definition(contd):

It is observed that a change in head orientation in the form of looking or glancing at the traffic is a strong indicator of crossing intention.

Our project aims to aid the interaction between the autonomous vehicles and the pedestrians. Overall, our work formulates the problem of pedestrian-driver interaction and sheds light on its complexity in typical traffic scenarios.

User Interfaces

UI-1: The system must recognize and respond to potential, immediate, and imminent hazardous situations involving collisions with pedestrians.

UI-2: Safety Effectiveness

2.1 The system shall detect and respond with a 100% effectiveness in regard to the supplied scenarios defined in the specification.

2.2 Safety can be computed by comparing the vehicle's collision rate to the population's "normal" collision rate.

2.3 When the system starts, a visual and audio alert should be given to the driver

2.4 The driver must have override capability (by pressing the brake)

2.5 There must not be external warnings that would alter the behavior of the pedestrian.

UI-3: Efficiency

3.1 The system will attempt to return to a steady state velocity as soon as the potential hazard is no longer seen as a threat.

3.2 Safety is always our number one priority, efficiency is always seen as a secondary concern.

UI-4: Pedestrian

4.1 The pedestrian may be motionless or in motion, with velocity of 6 kph.

4.2 Assume that the pedestrian may change velocity with infinite acceleration.

4.3 Consider the pedestrian to be a circle with a .5m diameter.

4.4 When the pedestrian is in motion, they may only move with right angles relative to the vehicle path.

UI-5: Fail Safes

5.1 If the brake system is in fail operational mode, the response time to reach the requested deceleration changes from 200ms to 900ms.

5.2 The algorithm should adjust to maintain zero collisions in trade for increased lost time.

Hardware Requirements

HI-1: 8GB of computer RAM alongside a minimum amount (128GB) of secondary storage space

Software Requirements

SI-1: Operating system-Windows 7 or above, Linux(4.2.x) or above.

Communication Requirements

CI-1: The system shall send information to the self driven vehicle regarding the pedestrians in front of the vehicle.

Other Non-Functional Requirements

NFR-1: The system shall be highly efficient in terms of processing the video frames and analyzing.

NFR-2: The system sensors shall cover the entire range of the path.

Performance Requirements

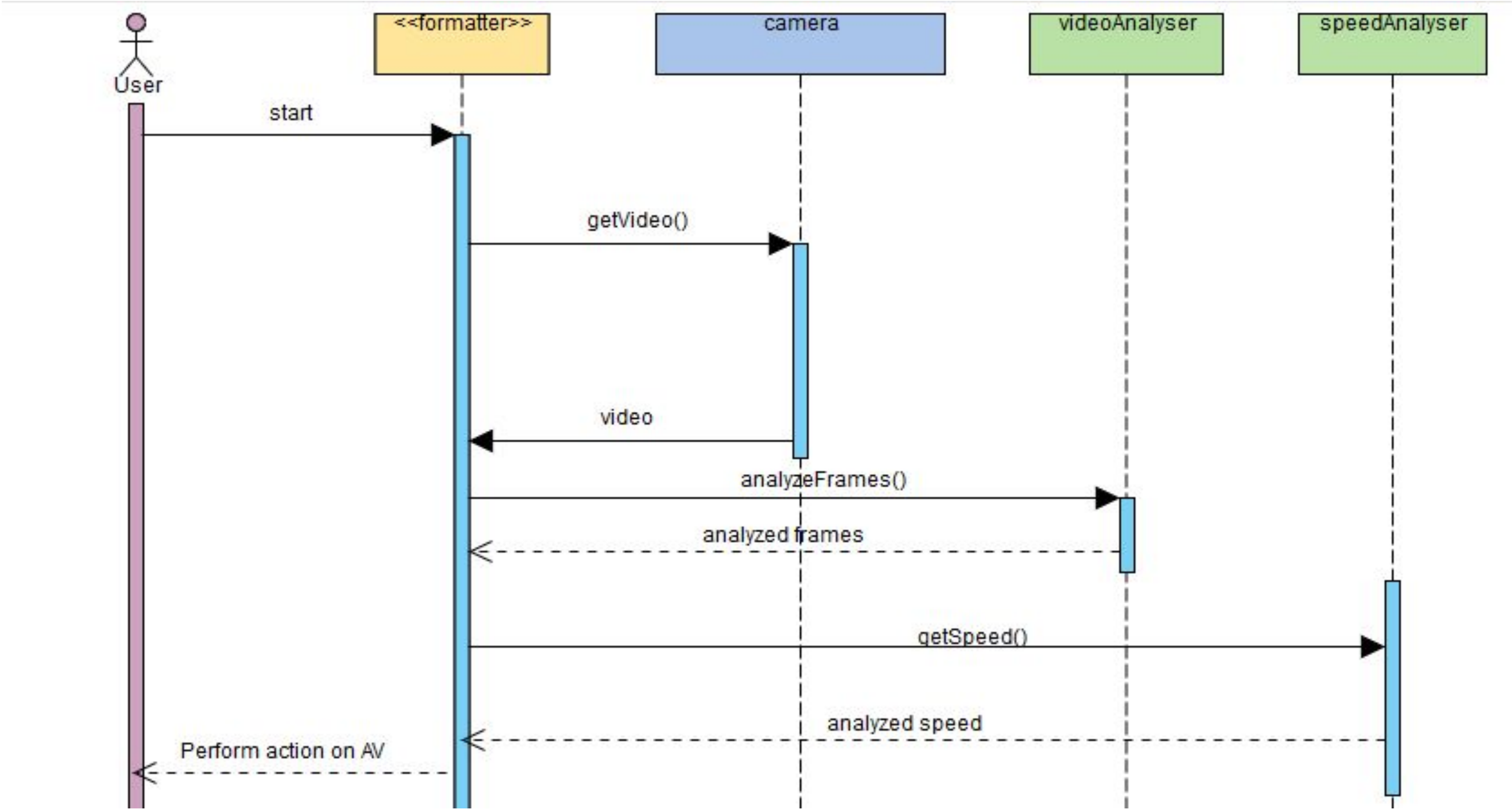
PE-1: Scalability: The system can be scaled up as it is individual to the vehicle

PE-2: Platform considerations: The system works on Linux and Windows environments

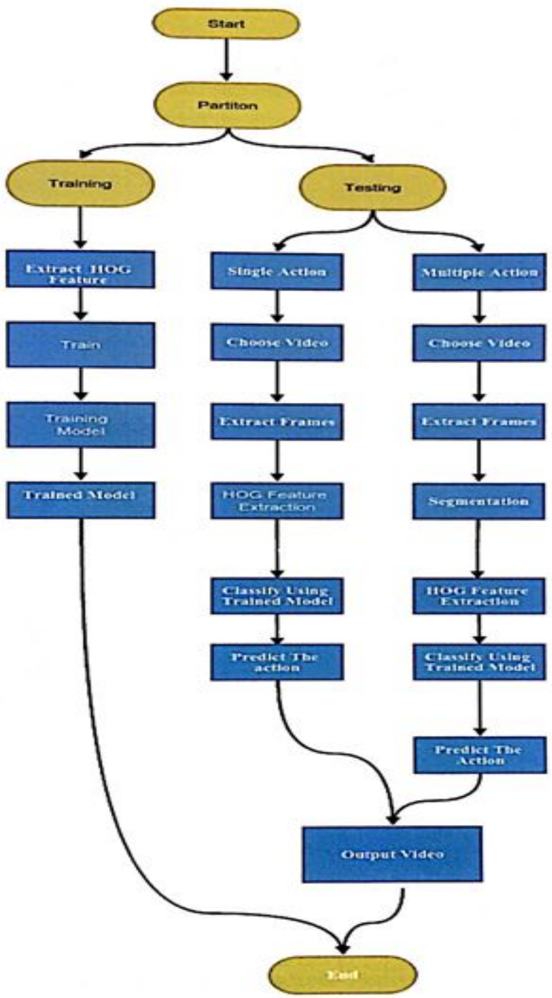
Safety Requirements

The system ensures safety of the vehicle, passengers as well as the pedestrians as the fail safes are implemented properly.

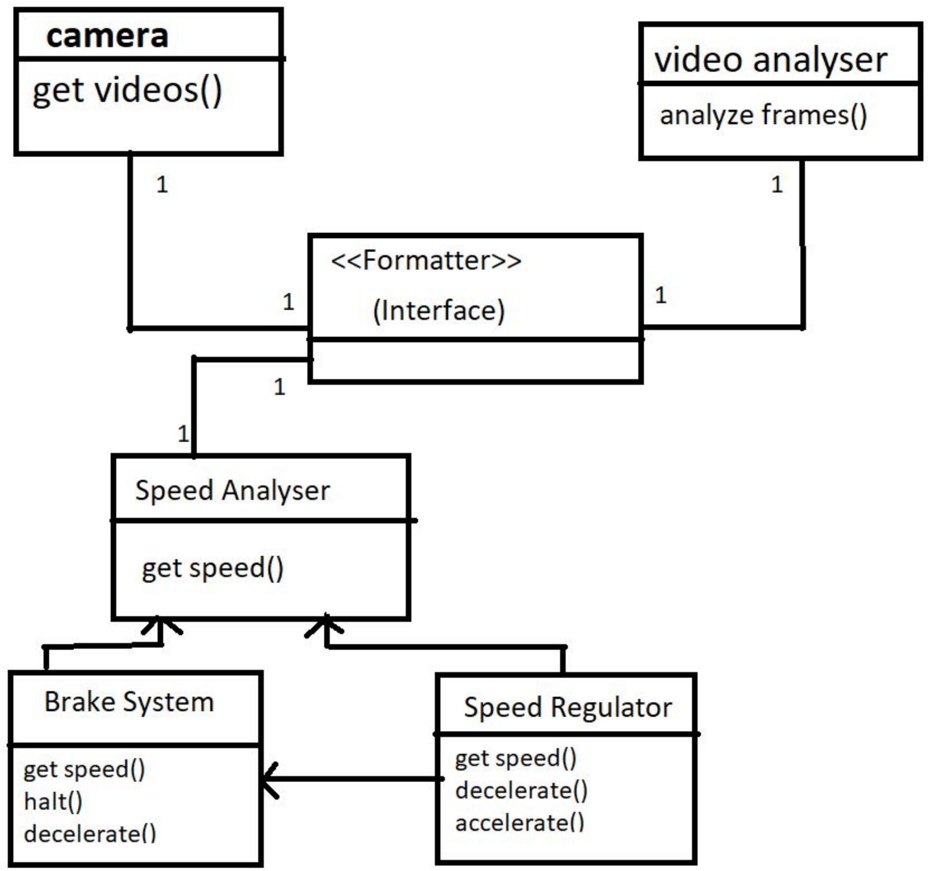
Sequence Diagram



SYSTEM ARCHITECTURE



Detailed design



Algorithm:

MTCNN (Multi-task Cascaded Convolutional Networks)

Step 1: Identifying the faces.
It predicts potential face positions and their bounding boxes like an attention network in **Faster R-CNN**. The result of this step is a large number of face detections and lots of false detections.

The second part uses images and outputs of the first prediction. It makes a refinement of the result to eliminate most of false detections and aggregate bounding boxes.

The last part refines even more the predictions and adds facial landmarks predictions (in the original MTCNN implementation).

Step 2: Cropping the faces
The bounding boxes of the images are taken and using those four points the faces in the image are cropped.

Step 3: Clustering them together.
Clustering is a process of partitioning a set of data(here images) into a set of meaningful sub-classes, called cluster.
The cropped images are used for clustering and image embeddings are created.

Step 4: Removing Noise.

During the process of face cropping and alignment, a lot of false faces are generated. These false faces need to be deleted to improve the overall workflow.

Step 5: Labeling Images.

When repeatedly doing step 4, we start getting the similar faces in same clusters. By the end of this step we end up with many folders inside which we will have images of that particular person.

Step 6: Making a classifier.

Using a folder name as the label and images as the example inputs, we make a classifier.

Development environment :

- Algorithm :MTCNN
- Editor:Pycharm
- OpenCV
- TensorFlow

Implementation approach

The MTCNN model consists of 3 separate networks: the P-Net, the R-Net, and the O-Net:
For every image we pass in, the network creates an image pyramid: that is, it creates multiple copies of that image in different sizes.

In the P-Net, for each scaled image, a 12x12 kernel runs through the image, searching for a face.The red square represents the kernel, which slowly moves across and down the image, searching for a face.

Within each of these 12x12 kernels, 3 convolutions are run through with 3x3 kernels. After every convolution layer, a prelu layer is implemented. In addition, a maxpool layer is put in after the first prelu layer(maxpool takes out every other pixel, leaving only the largest one in the vicinity).

R-Net has a similar structure, but with even more layers. It takes the P-Net bounding boxes as its inputs, and refines its coordinates.

Similarly, R-Net splits into two layers in the end, giving out two outputs: the coordinates of the new bounding boxes and the machine's confidence in each bounding box.

Finally, O-Net takes the R-Net bounding boxes as inputs and marks down the coordinates of facial landmarks.
O-Net splits into 3 layers in the end, giving out 3 different outputs: the probability of a face being in the box, the coordinates of the bounding box, and the coordinates of the facial landmarks (locations of the eyes, nose, and mouth).

Testing strategy and Vaildation Procedure

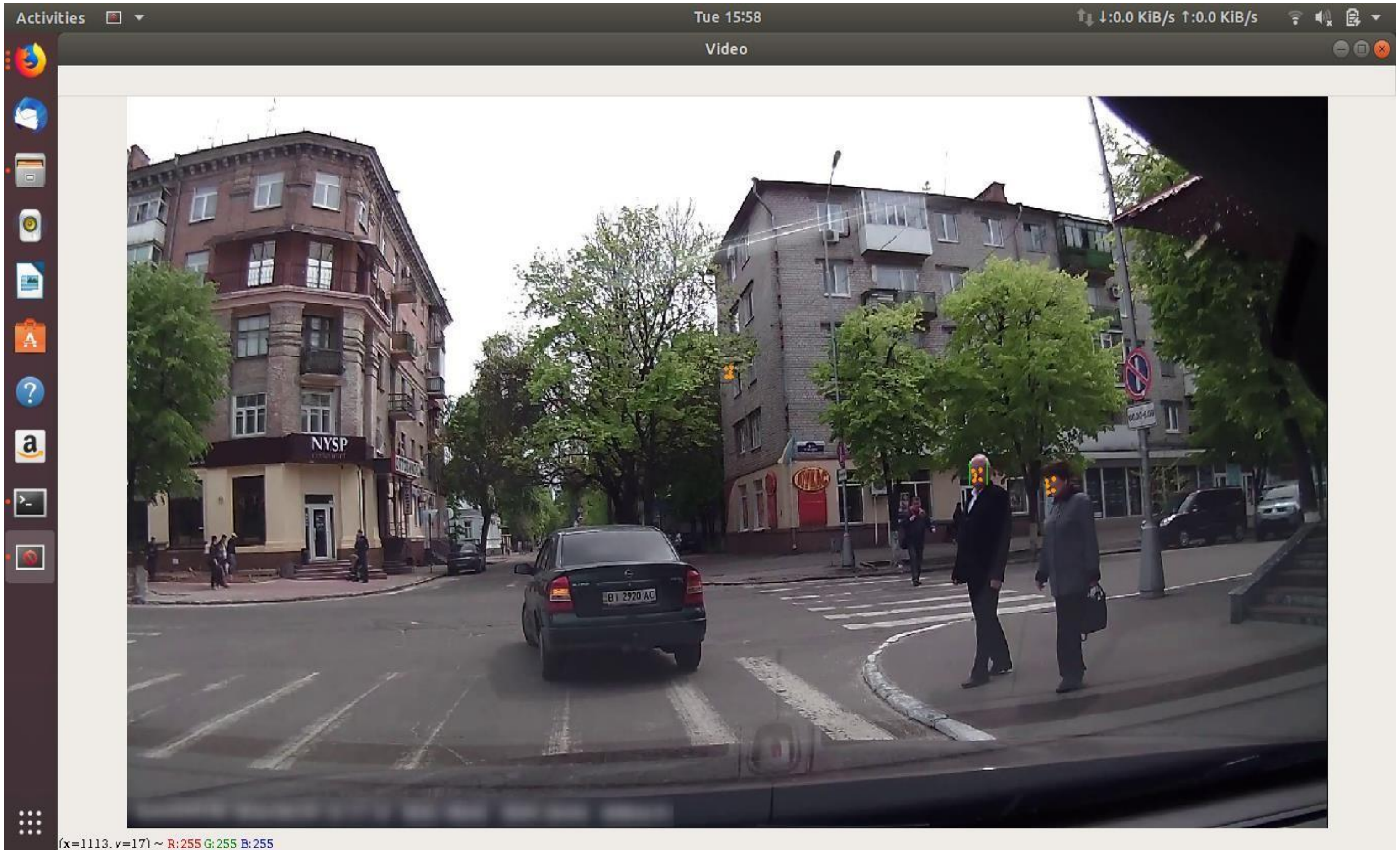
The testing of the project is done using 25 different videos in different scenarios, weather conditions, number of pedestrians, street view etc. The results of the tests are tabulated and accuracy is mentioned.

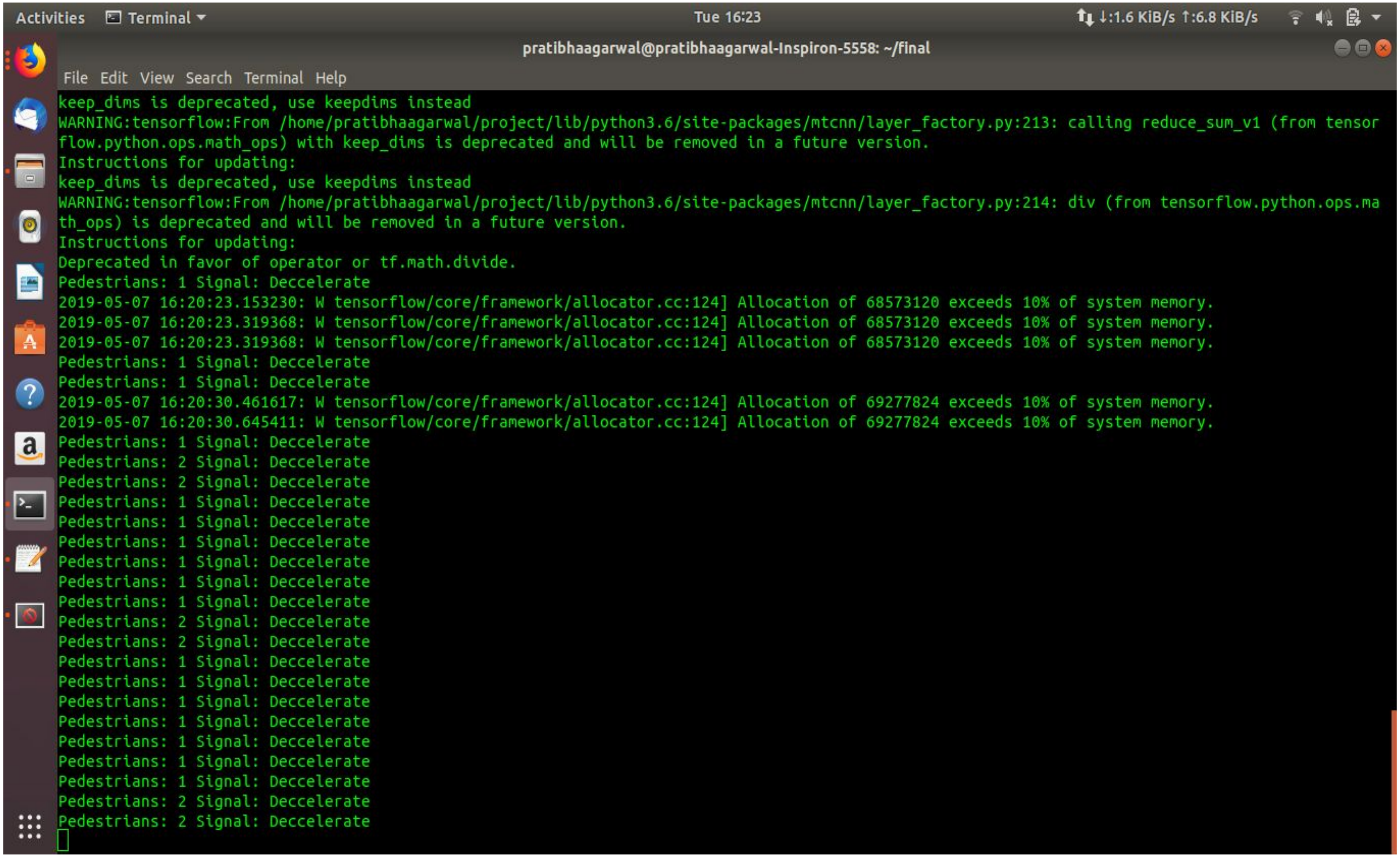
The tests show that the system can detect pedestrians with an average accuracy of 91%. Thus the signals are sent to the autonomous vehicle

Understanding Pedestrian Behaviour to Improve working of Autonomous Vehicles

Video Sample	No. of Pedestrians	Weather	No. of Lanes	No. of Pedestrians detected	Accuracy(%)
Video 1	2	Clear	2	2	100
Video 2	3	Cloudy	2	3	100
Video 3	4	Clear	1	3	75
Video 4	5	Rain	3	5	100
Video 5	5	Clear	4	4	80
Video 6	6	Cloudy	3	5	84
Video 7	6	Snow	2	6	100
Video 8	6	Rain	3	5	84
Video 9	7	Clear	4	7	100
Video 10	7	Cloudy	3	6	86
Video 11	7	Snow	4	5	72
Video 12	8	Clear	2	8	100
Video 13	8	Cloudy	3	7	88
Video 14	8	Rain	5	6	75
Video 15	9	Clear	4	9	100
Video 16	9	Snow	3	8	89
Video 17	9	Cloudy	2	9	100
Video 18	10	Clear	3	10	100
Video 19	10	Rain	4	9	90
Video 20	10	Snow	2	9	90
Video 21	10	Cloudy	3	10	100
Video 22	11	Clear	2	11	100
Video 23	11	Snow	3	9	82
Video 24	11	Cloudy	4	11	100
Video 25	11	Rain	3	10	91

RESULTS


Activities
Tue 15:58
↓:0.0 KiB/s ↑:0.0 KiB/s
Video
NYSP
(x=1113, y=17) ~ R:255 G:255 B:255


Activities
Terminal
Tue 16:23
↓:1.6 KiB/s ↑:6.8 KiB/s
pratibhaagarwal@pratibhaagarwal-Inspiron-5558: ~/final
File Edit View Search Terminal Help
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From /home/pratibhaagarwal/project/lib/python3.6/site-packages/mtcnn/layer_factory.py:213: calling reduce_sum_v1 (from tensorflow.python.ops.math_ops) with keep_dims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From /home/pratibhaagarwal/project/lib/python3.6/site-packages/mtcnn/layer_factory.py:214: div (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Pedestrians: 1 Signal: Deccelerate
2019-05-07 16:20:23.153230: W tensorflow/core/framework/allocator.cc:124] Allocation of 68573120 exceeds 10% of system memory.
2019-05-07 16:20:23.319368: W tensorflow/core/framework/allocator.cc:124] Allocation of 68573120 exceeds 10% of system memory.
2019-05-07 16:20:23.319368: W tensorflow/core/framework/allocator.cc:124] Allocation of 68573120 exceeds 10% of system memory.
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
2019-05-07 16:20:30.461617: W tensorflow/core/framework/allocator.cc:124] Allocation of 69277824 exceeds 10% of system memory.
2019-05-07 16:20:30.645411: W tensorflow/core/framework/allocator.cc:124] Allocation of 69277824 exceeds 10% of system memory.
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 1 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate
Pedestrians: 2 Signal: Deccelerate

REFERENCES

1. Amir Rasouli , Iuliia Kotseruba , and John K. Tsotsos, “Understanding Pedestrian Behavior in Complex”
Traffic Scenes
The study shows that changes in head orientation in the form of looking or glancing
at the traffic is a strong indicator of crossing intention.

2. T. M. Gasser, “Fundamental and special legal questions for autonomous vehicles,” Autonomous Driving, Technical, Legal and Social Aspects, Berlin, Germany: Springer-Verlag, 2016, pp. 523–551
It looks at degrees of automation that will become technically feasible in the distant future.

3. I. Wolf, “The interaction between humans and autonomous agents,” in Autonomous Driving. Berlin, Germany: Springer-Verlag, pp. 103–124, 2016
It uses cognitive psychology as a central theoretical paradigm for understanding and designing the interaction between humans and technical system

4. B. Farber, “Communication and communication problems between autonomous vehicles and human drivers,” in Autonomous Driving. Berlin, Germany: Springer-Verlag, 2016, pp. 125–144.
It discusses the various facets of communication between the autonomous

Thank you