Department of Information Science and Technology

IT7811 - PROJECT WORK

Project Title	-	**Real Time Driver Drowsiness detection**	
Project Guide	-	Dr. Abirami Murugappan	
Project Members	-	Danush S	- 2017115518
		Lalith Prasad S	- 2017115544
		Premchander S	- 2017115572

Introduction

- ❖ In this project, we aim to develop Deep Learning models (Computer vision) that are capable of detecting drowsiness of driver based on facial features.
- ❖ The implementation of the models is done on a Raspberry Pi, so it can work as a standalone system and capture input as well as make predictions without requiring external computational resources.
- ❖ This project aims to prevent accidents that are caused due to driver fatigue.

Problem statement

- The existing systems mostly depend on a cloud based environment. The proposed system is built using raspberry pi, which enables it to function completely in a local environment, which makes it suitable for real time use.
- The proposed system uses computer vision in order to achieve the expected output, so it does not require any sensors to be in contact with the driver. In contact approaches, drivers wear or touch some devices to get physiological parameters for detecting the level of their fatigue.
- Most of the accidents due to fatigue happen during night time. This system aims to perform with high accuracy even in low light conditions.

Objectives

- ❖ Hardware implementation using Raspberry Pi with real time drowsiness detection.
- ❖ Predict whether the driver is going to fall asleep in a few seconds using key facial features and forewarn driver before he falls asleep.
- ❖ Time series prediction of the next day’s timings when the driver may fall asleep based on previous drowsiness history.
- ❖ Out of frame inspection and deviation ratio to suggest camera tilt direction whenever driver goes out of frame coverage.
- ❖ Luminance boosting of low light frames to act as better input to the models.

Literature survey

S.NO	PAPER	DESCRIPTION	ALGORITHMS USED	LIMITATION
[1]	Nashwan Adnan Othman, Mehmet Umut Salur, "**An Embedded Real-Time Object Detection and Measurement of its Size**", International Conference on Artificial Intelligence and Data Processing (IDAP), 2018	An enhanced technique for detecting objects and computing their measurements in real time from video streams.	OpenCV, canny edge detection, erosion algorithm	The accuracy isn't perfect since this is due to the seeing angle and lens deformation.

Literature survey

S.NO	PAPER	DESCRIPTION	ALGORITHMS USED	LIMITATION
[2]	Suchitra,Suja.P, Shikha Tripathi, “**Real-Time Emotion Recognition from Facial Images using Raspberry Pi II”,** 3rd International Conference on Signal Processing and Integrated Networks (SPIN), 2016	A system which detects real time emotion recognition from facial image.Emotions can understood by text, vocal, verbal and facial expressions.	Haar Cascade, AdaBoost Classifier.	More work need to be done on images with different illumination conditions as in real time these global conditions are not uniform.

Literature survey

S.NO	PAPER	DESCRIPTION	ALGORITHMS USED	LIMITATION
[3]	Sreelekshmy Selvin , R. Vinayakumar, E.A. Gopalakrishna, Vijay Krishna Menon, and K.P. Soman. **"Stock price prediction using LSTM, RNN and CNN-sliding window model".**International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.	Deep learning architectures(LSTM ,RNN and CNN-sliding window) for predicting future price values on a short term basis.	LSTM,RNN,CNN	Stock market is a highly dynamical system, the patterns and dynamics existing with in the system will not always be the same.This cause learning problems to LSTM and RNN architecture and hence the models fails to capture the dynamical changes accurately.

Literature survey

S.NO	PAPER	DESCRIPTION	ALGORITHMS USED	LIMITATION
[4]	Yi Sun , Xiaogang Wang, and Xiaoou Tang. **”Deep Convolutional Network Cascade for Facial Point Detection”**.IEEE Conference on Computer Vision and Pattern Recognition, 2013.	Estimation of the positions of facial keypoints with three-level convolutional networks. At each level, the outputs of multiple networks are fused for robust and accurate estimation.	CNN, Multi-level Regression.	Facial keypoint detection causes problem when face images are taken with extreme poses, lightings, expressions, and occlusions.

Module description

1. Luminance boosting

Input: Segmented frames from the camera’s video stream.

Output: Frame with increased brightness

- First, the brightness of the frame is checked by converting the color space of the frame from bgr to ycbcr
- The brightness of the frame is then calculated from the y channel which represents the luma characteristic of the image
- If low brightness is detected, the b, g and r channels of the frame are equalized and the brightness of the frame is increased

Module description

2. Drowsiness detection

Input: Segmented frames from the camera's video stream.

Output: Binary output indicating whether the driver has fallen asleep or not

- Live video stream is captured and separated as individual frames
- Each frame first goes through face detection (HOG + SVM) model which outputs coordinates of region covering the face in the frame.
- Then, the frame goes through facial landmarks detection (Ensemble of regression trees) model which outputs the coordinates of eyes and mouth in the frame.
- Aspect ratios of eyes and mouth are then separately calculated, based on which the state of the driver - drowsy or not, can be predicted.

Module description

3. Drowsiness prediction

Input: Segmented frames from the camera's video stream.

Output: Binary output indicating whether the driver is going to fall asleep or not

- From the video stream, every 15th frame is chosen as input
- The model is trained with two classes of images indicating - going to fall asleep and awake
- Each image in the dataset is a picture of face representing the corresponding class
- The model then predicts which class the frame belongs to and alerts if the class is 'going to fall asleep', indicating that the driver is going to fall asleep.

Module description

4. Sleep prediction

Input: Timestamps of instances when drowsiness was detected

Output: Timestamps for the next day when the driver may fall asleep

- Whenever drowsiness is detected, the timestamps are recorded and uploaded to the firebase database
- This set of timestamps, representing the drowsiness history of a specific driver, forms the dataset for the model
- The timestamps are then preprocessed as suitable input to the LSTM model
- The LSTM model then predicts the timings during next day at which the driver may fall asleep

Module description

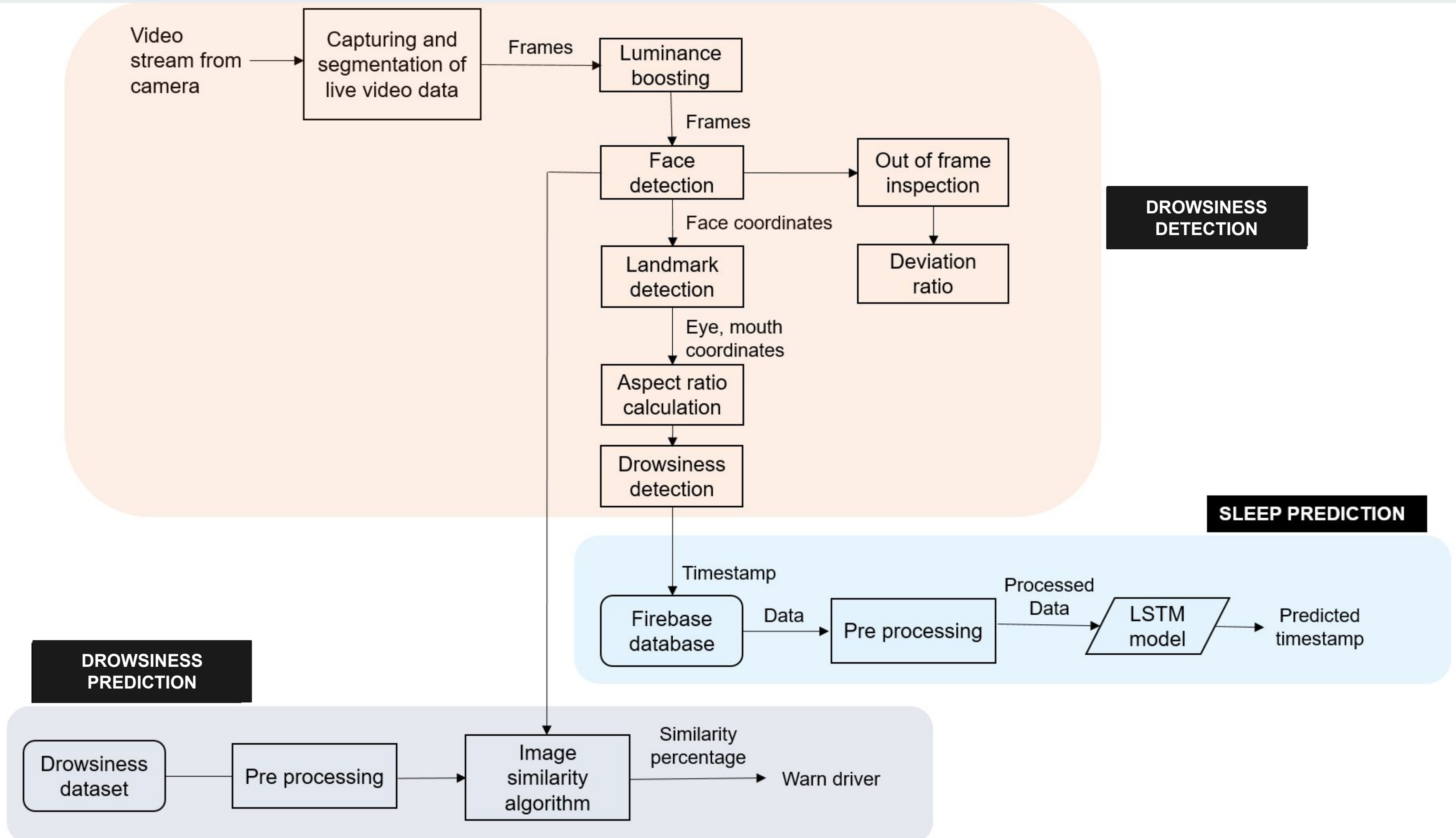
5. Out of frame inspection

Input: Coordinates of face in the frame

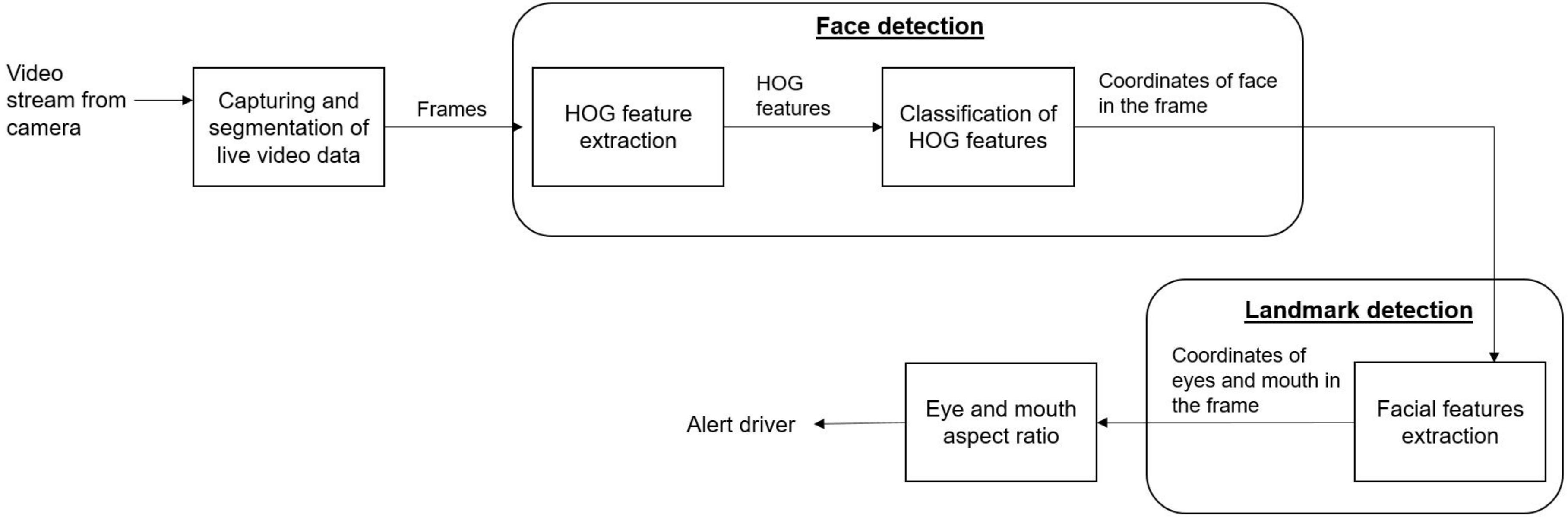
Output: Suggested direction to tilt the camera

- Using the coordinates, the midpoint (x,y) of face is calculated
- A global variable keeps track of the last recorded location of the face
- Based on the location of the midpoint in the frame, the variable is updated for each consecutive frame as left, right, bottom or top
- For every frame in which the face is not detected, the value in the variable is given as suggestion of direction to tilt the camera
- Deviation ratio is the ratio of number of frames which doesn’t contain a face to number of frames containing a face - this is used as a measure of how well the system is set up and whether it requires any changes.

Overall System Architecture

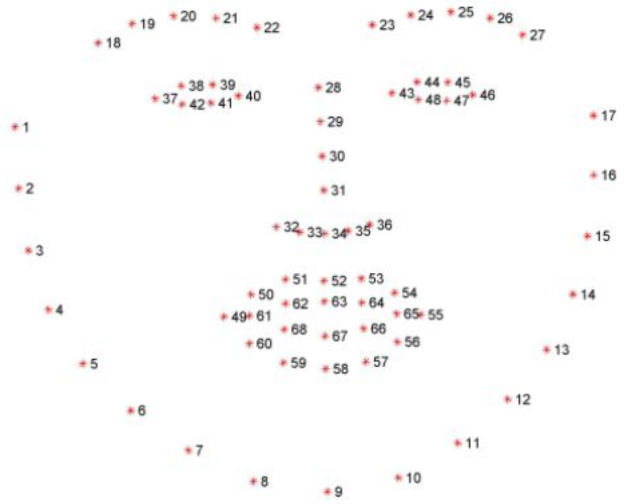


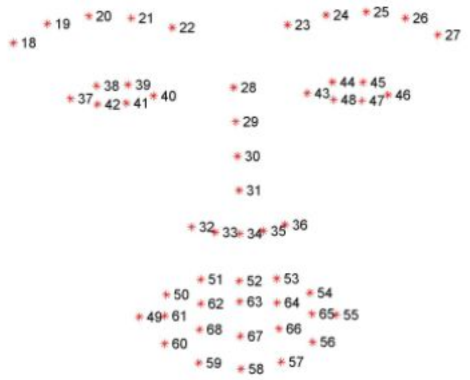
Block diagram - Drowsiness detection


Video stream from camera
Capturing and segmentation of live video data
Frames
Face detection
HOG feature extraction
HOG features
Classification of HOG features
Coordinates of face in the frame
Landmark detection
Facial features extraction
Coordinates of eyes and mouth in the frame
Eye and mouth aspect ratio
Alert driver

Dataset - Drowsiness detection

- Source - ibug.doc.ic.ac.uk - Faces In-the-Wild Challenge
- The dataset consists of images of face along with 68 points of annotation for each image

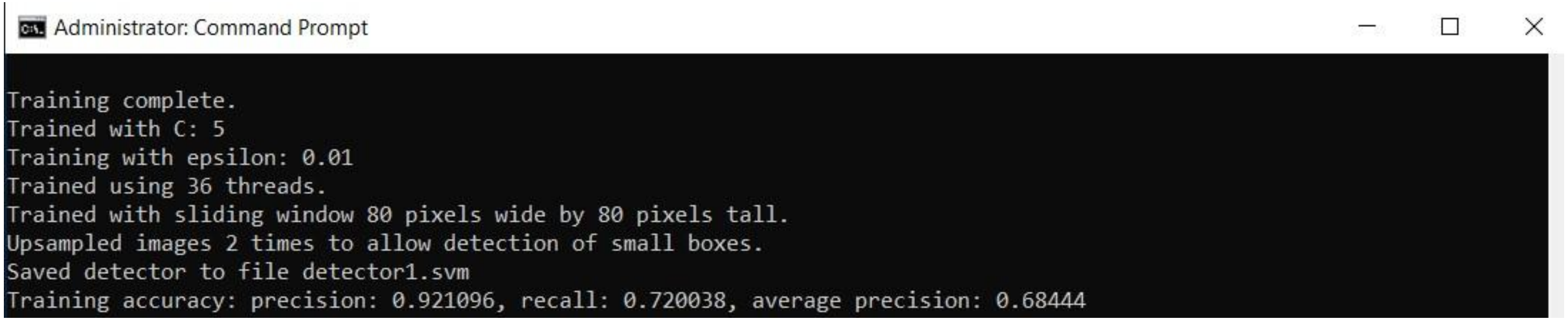




Results - Drowsiness detection

Face detection model

Training accuracy: 92.10%

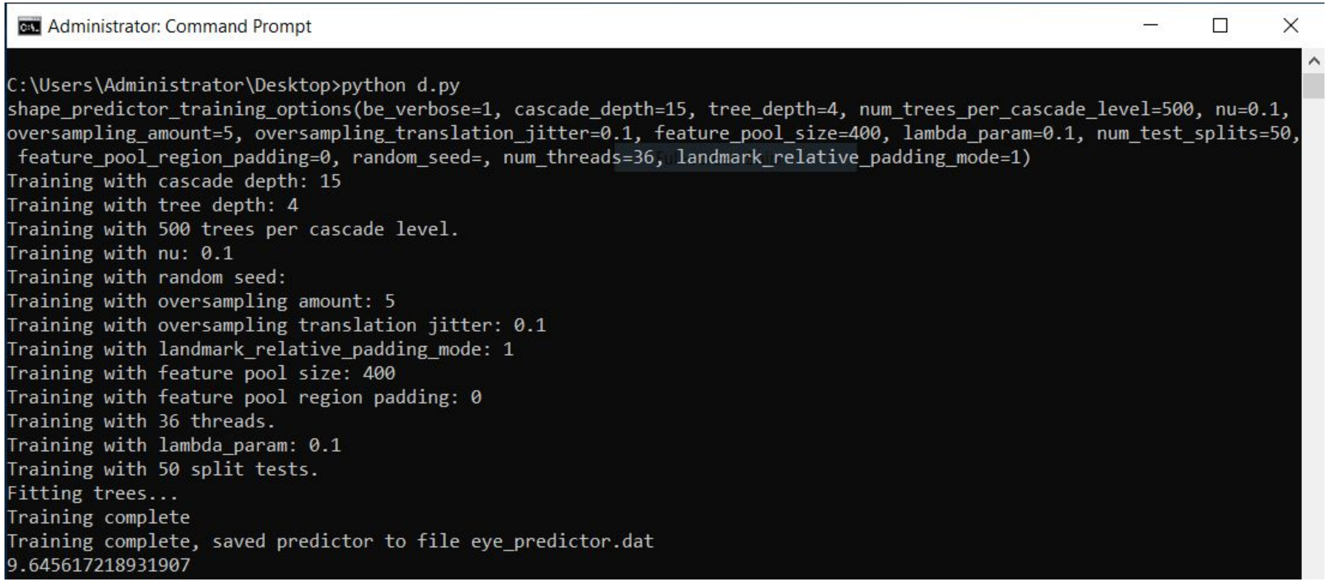


```
Training complete.
Trained with C: 5
Training with epsilon: 0.01
Trained using 36 threads.
Trained with sliding window 80 pixels wide by 80 pixels tall.
Upsampled images 2 times to allow detection of small boxes.
Saved detector to file detector1.svm
Training accuracy: precision: 0.921096, recall: 0.720038, average precision: 0.68444
```

Results - Drowsiness detection

Facial landmarks detection model

Training accuracy: 96.4%

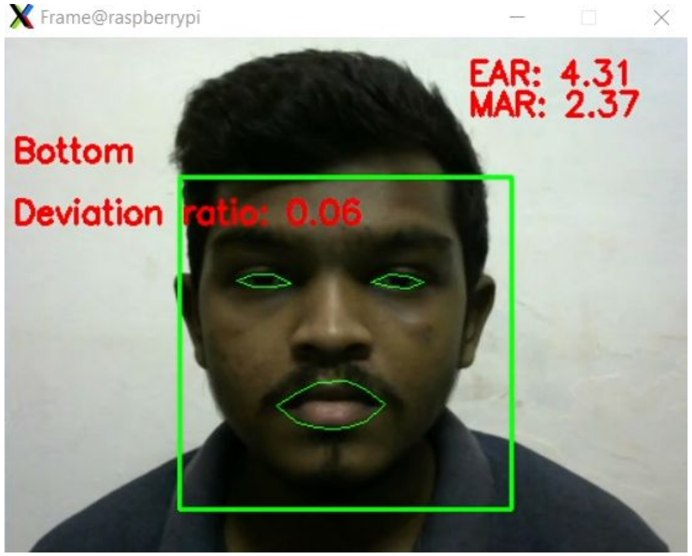


Administrator: Command Prompt

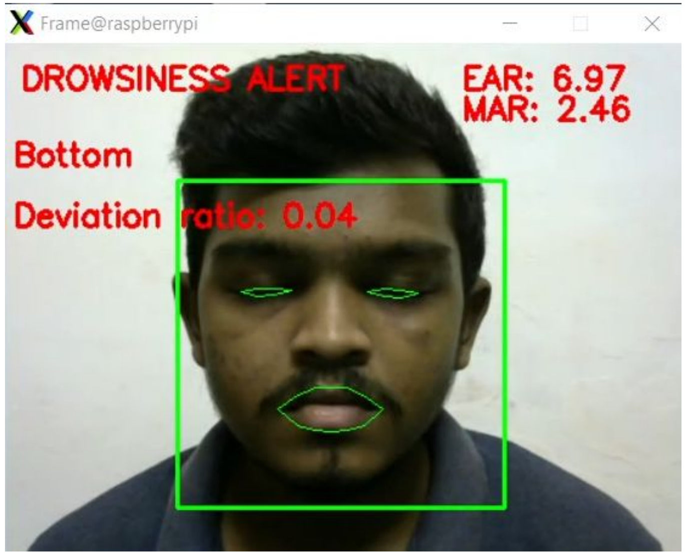
```
C:\Users\Administrator\Desktop>python d.py
shape_predictor_training_options(be_verbose=1, cascade_depth=15, tree_depth=4, num_trees_per_cascade_level=500, nu=0.1,
oversampling_amount=5, oversampling_translation_jitter=0.1, feature_pool_size=400, lambda_param=0.1, num_test_splits=50,
 feature_pool_region_padding=0, random_seed=, num_threads=36, landmark_relative_padding_mode=1)
Training with cascade depth: 15
Training with tree depth: 4
Training with 500 trees per cascade level.
Training with nu: 0.1
Training with random seed:
Training with oversampling amount: 5
Training with oversampling translation jitter: 0.1
Training with landmark_relative_padding_mode: 1
Training with feature pool size: 400
Training with feature pool region padding: 0
Training with 36 threads.
Training with lambda_param: 0.1
Training with 50 split tests.
Fitting trees...
Training complete
Training complete, saved predictor to file eye_predictor.dat
9.645617218931907
```

Sample Output - Drowsiness detection

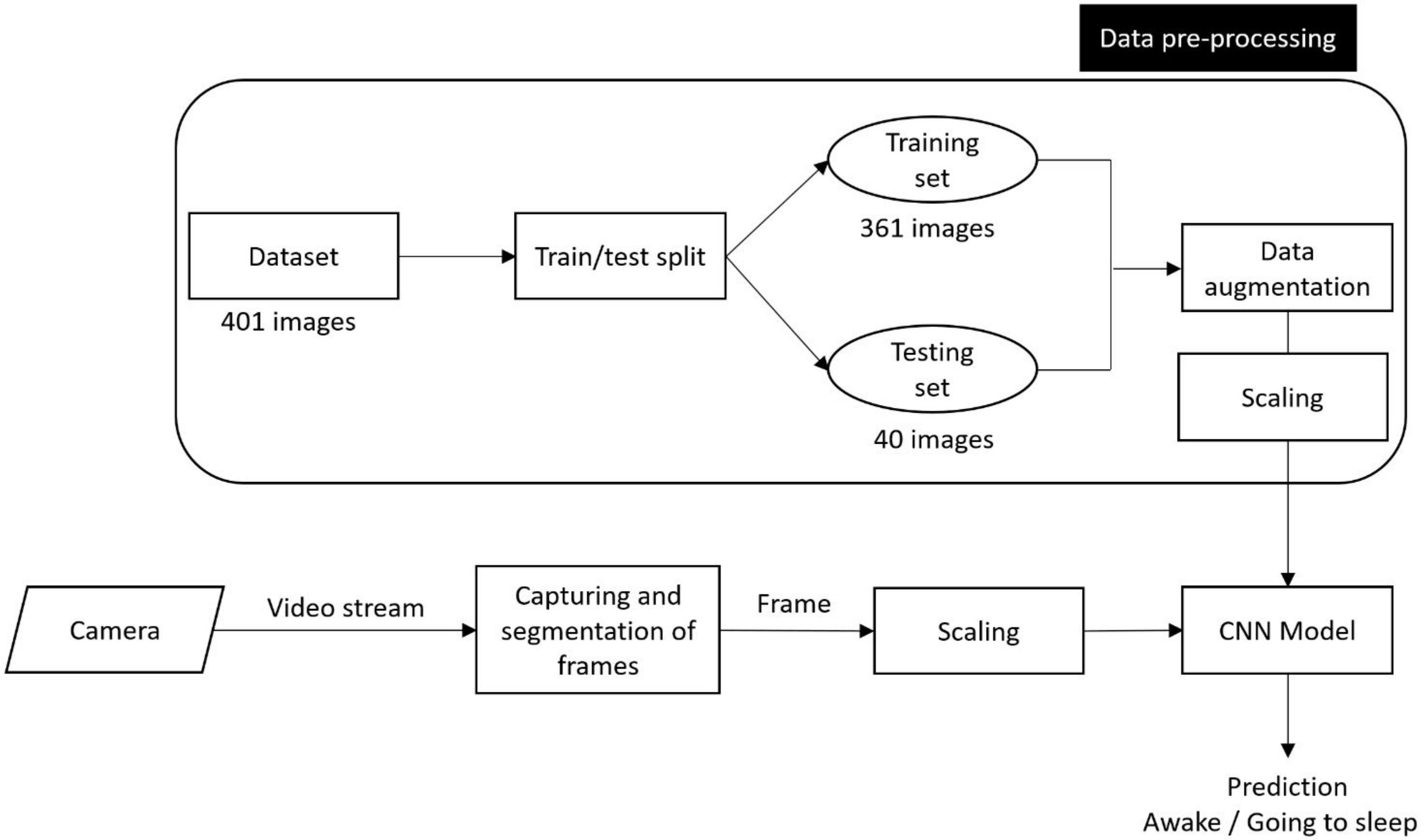
Not drowsy



Drowsy

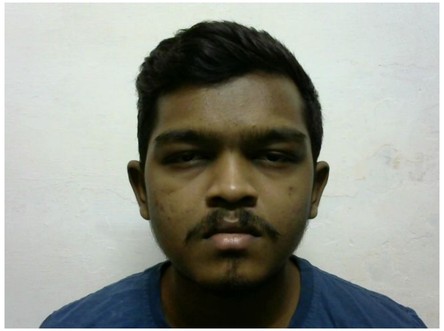


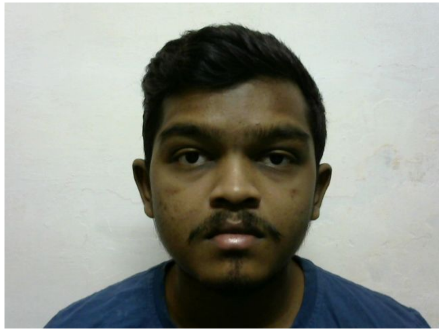
Block diagram - Drowsiness prediction



Dataset - Drowsiness prediction

- 401 images of face collected using webcam
- Split into 2 classes of images:
  - Awake - 124 images
  - Going to sleep - 277 images

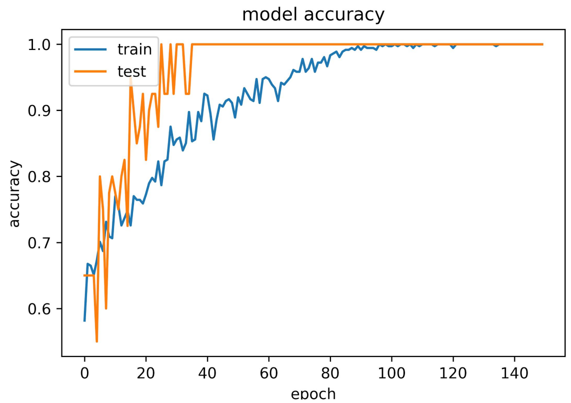

Going to sleep

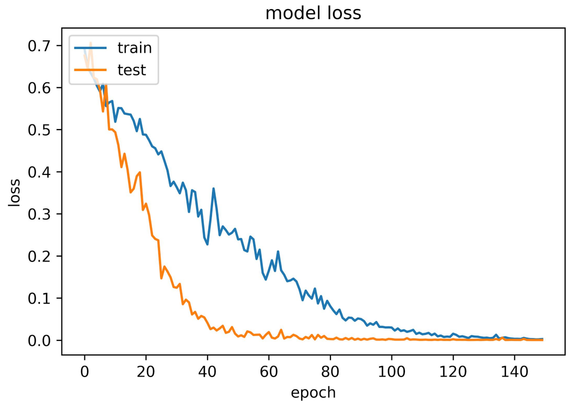

Awake

Pseudo code - Drowsiness prediction

- Load the saved CNN model
- Turn camera on and start video stream
- **<u>Image preprocessing:</u>**
  - Extract frame from stream
  - Convert color scheme from bgr to rgb
  - Reshape into 224,224 sized image, convert into array
  - Normalize all pixel values
- Process the input through the loaded model for prediction
- If the prediction infers that the driver is going to sleep, raise alert

Results - Drowsiness prediction





CNN Model

Training loss: 0.0027

Validation loss: 0.0003

Training, validation accuracy: 100%

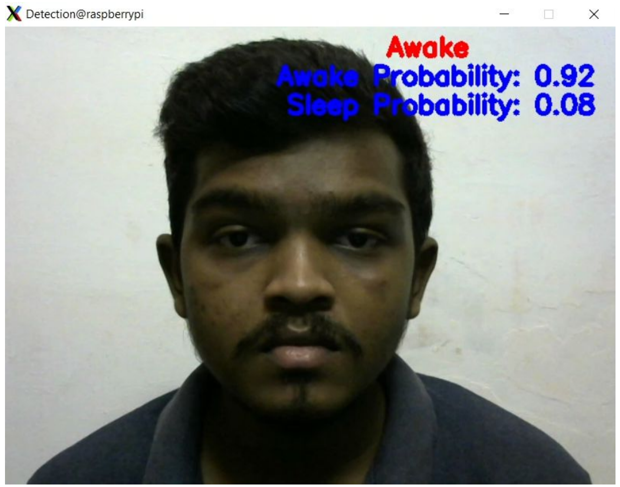
Results - Drowsiness prediction

Other models developed

MODEL	ACCURACY	LOSS
Transfer learning - Resnet	94%	0.44
Transfer learning - Mobilenet	93%	0.08
Siamese Network	80%	0.61

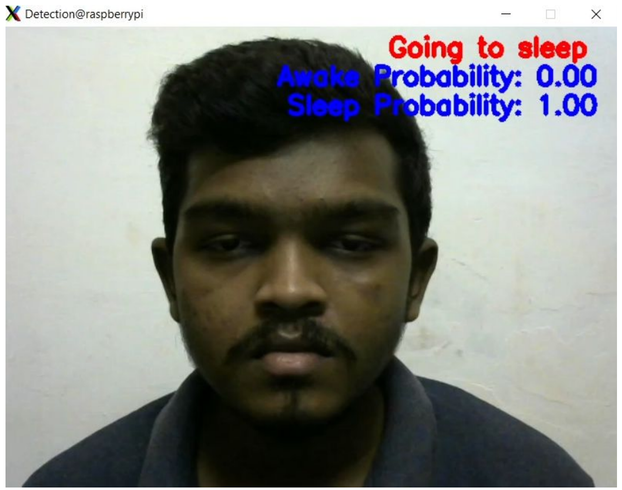
Sample Output - Drowsiness prediction

Awake



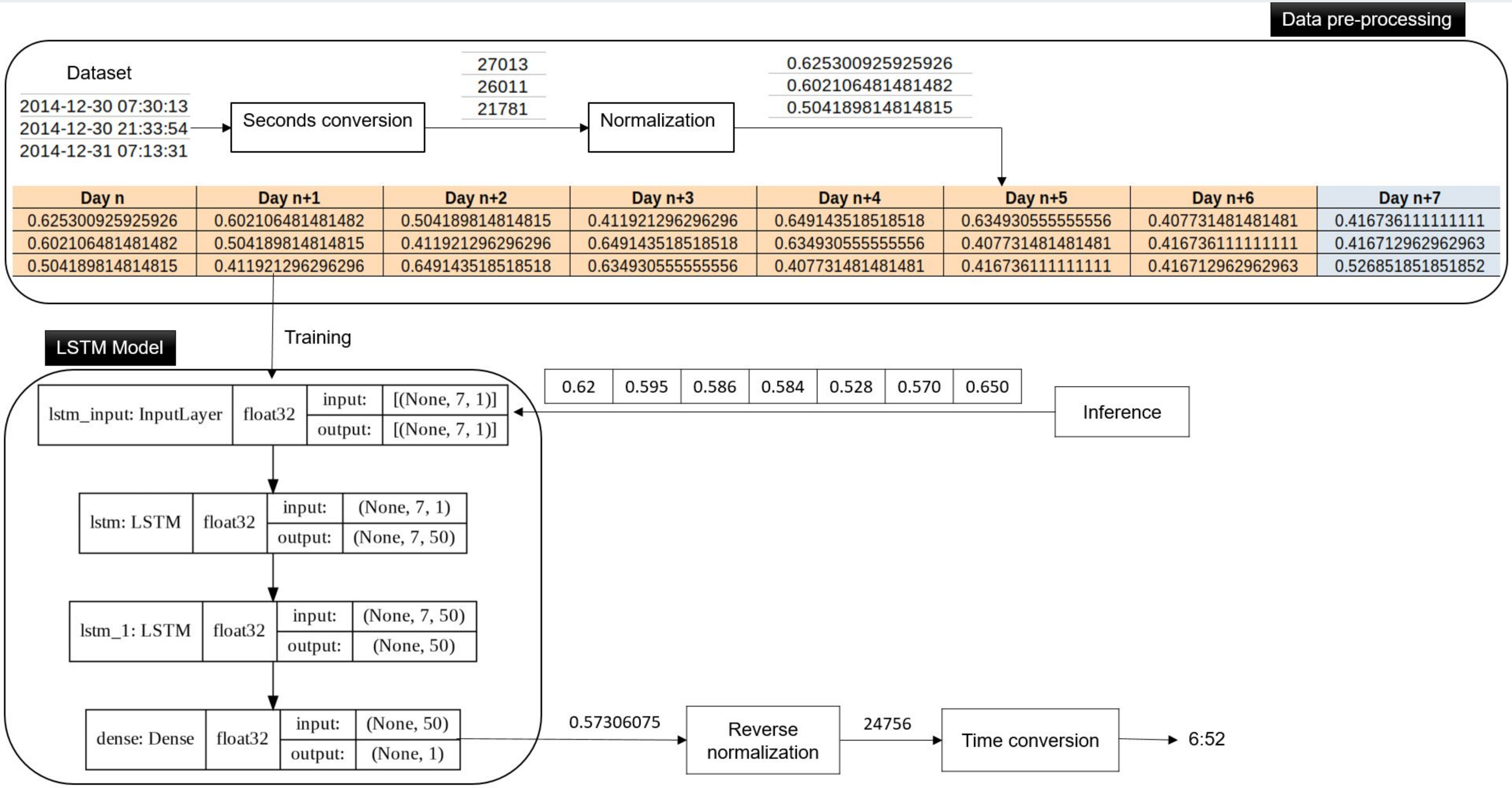
Awake
Awake Probability: 0.92
Sleep Probability: 0.08

Going to sleep



Going to sleep
Awake Probability: 0.00
Sleep Probability: 1.00

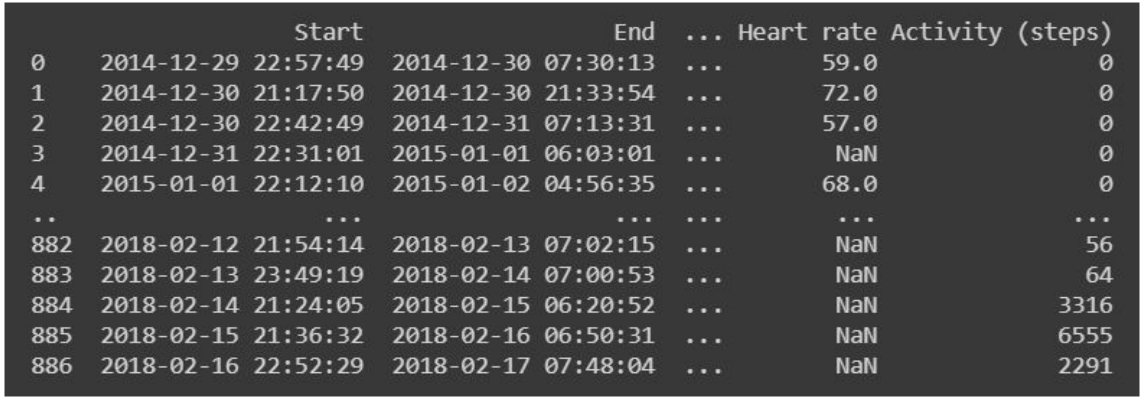
Block diagram - Sleep prediction



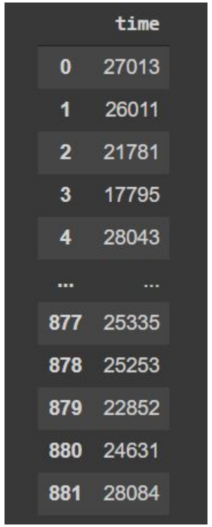
Day n	Day n+1	Day n+2	Day n+3	Day n+4	Day n+5	Day n+6	Day n+7
0.625300925925926	0.602106481481482	0.504189814814815	0.411921296296296	0.649143518518518	0.634930555555556	0.407731481481481	0.416736111111111
0.602106481481482	0.504189814814815	0.411921296296296	0.649143518518518	0.634930555555556	0.407731481481481	0.416736111111111	0.416712962962963
0.504189814814815	0.411921296296296	0.649143518518518	0.634930555555556	0.407731481481481	0.416736111111111	0.416712962962963	0.526851851851852

Dataset - Sleep prediction

- Source - Sleep data from Kaggle
- Number of data points - 887
- The dataset consists of timings of sleep recorded for an individual over 887 days



```
                   Start                  End  ... Heart rate Activity (steps)
0    2014-12-29 22:57:49  2014-12-30 07:30:13  ...       59.0                0
1    2014-12-30 21:17:50  2014-12-30 21:33:54  ...       72.0                0
2    2014-12-30 22:42:49  2014-12-31 07:13:31  ...       57.0                0
3    2014-12-31 22:31:01  2015-01-01 06:03:01  ...        NaN                0
4    2015-01-01 22:12:10  2015-01-02 04:56:35  ...       68.0                0
..                   ...                  ...  ...        ...              ...
882  2018-02-12 21:54:14  2018-02-13 07:02:15  ...        NaN               56
883  2018-02-13 23:49:19  2018-02-14 07:00:53  ...        NaN               64
884  2018-02-14 21:24:05  2018-02-15 06:20:52  ...        NaN             3316
885  2018-02-15 21:36:32  2018-02-16 06:50:31  ...        NaN             6555
886  2018-02-16 22:52:29  2018-02-17 07:48:04  ...        NaN             2291
```



	time
0	27013
1	26011
2	21781
3	17795
4	28043
...	...
877	25335
878	25253
879	22852
880	24631
881	28084

Before preprocessing

887 rows, 8 columns

After preprocessing

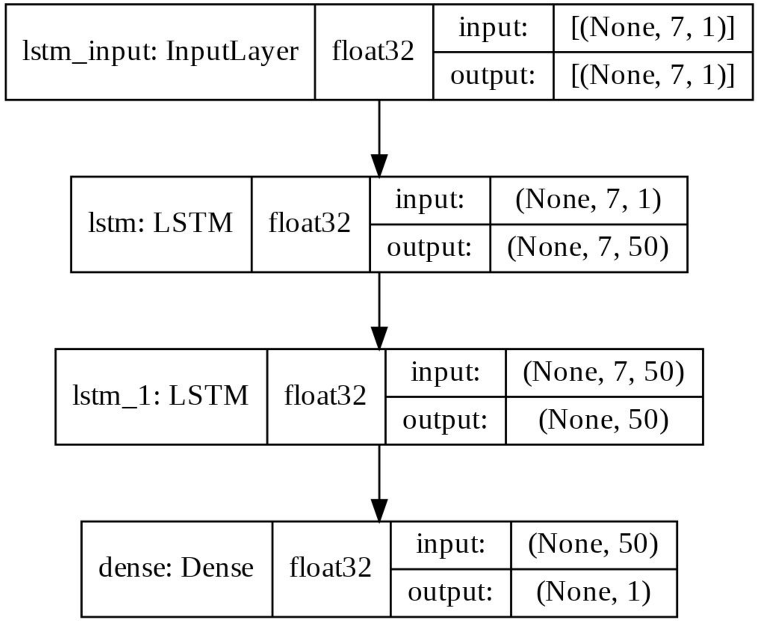
882 rows, 1 column

Pseudo code - Sleep prediction

- Extract latest entry from firebase
- Load the saved model
- <u>**Data preprocessing:**</u>
    - Convert timestamp to number of seconds
    - Normalize the value so that it falls between 0 and 1
    - Add the value to the existing list of previous days' history
    - Take the latest 7 values including new value and convert into array
- Process the input through the loaded model for prediction
- Reverse the normalization and obtain number of seconds
- Convert seconds to time, which is the predicted time during which the driver may fall asleep

Results - Sleep prediction

LSTM Model

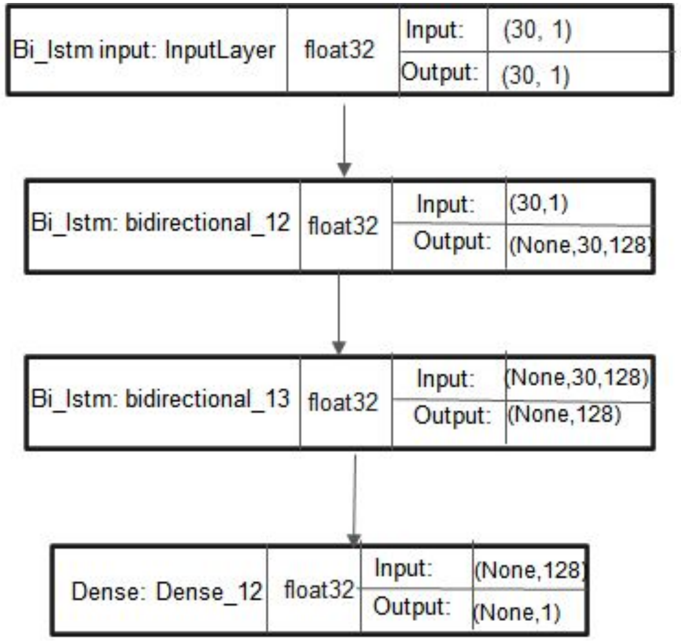


Loss type: Mean squared error

Loss: 0.005

Results - Sleep prediction

Bidirectional LSTM Model

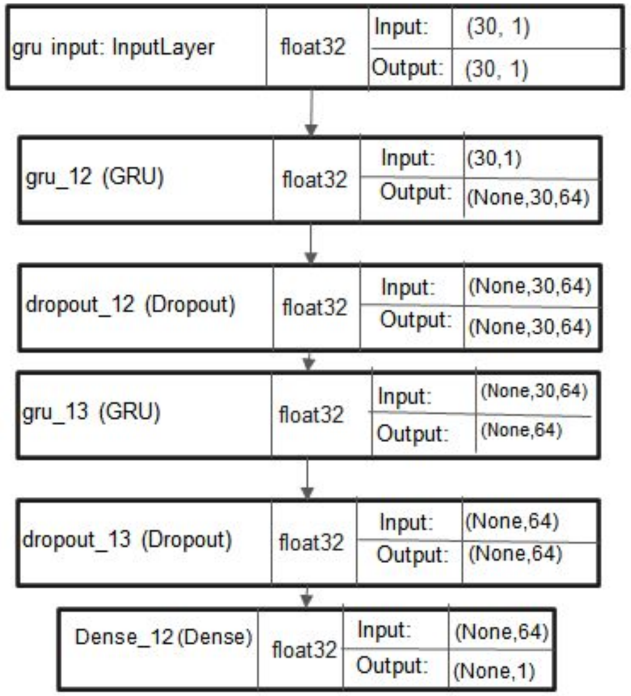


Loss type: Mean squared error

Loss: 0.0446

Results - Sleep prediction

GRU Model

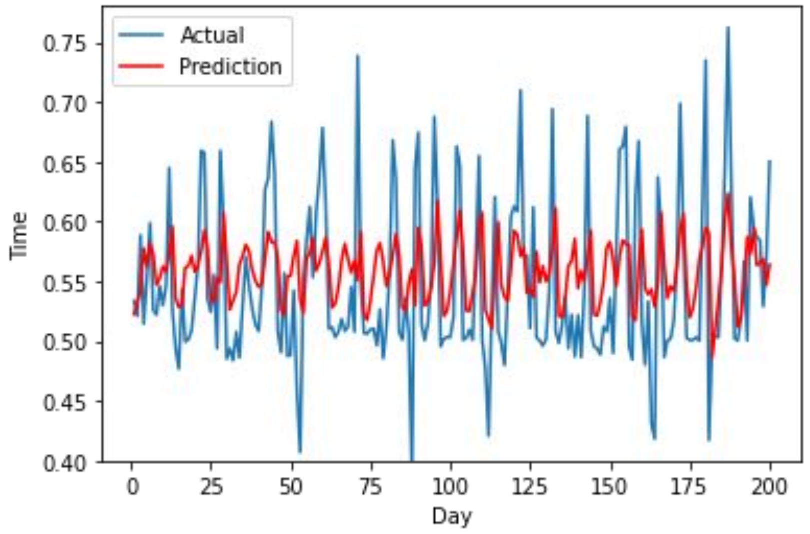


Loss type: Mean squared error

Loss: 0.0293

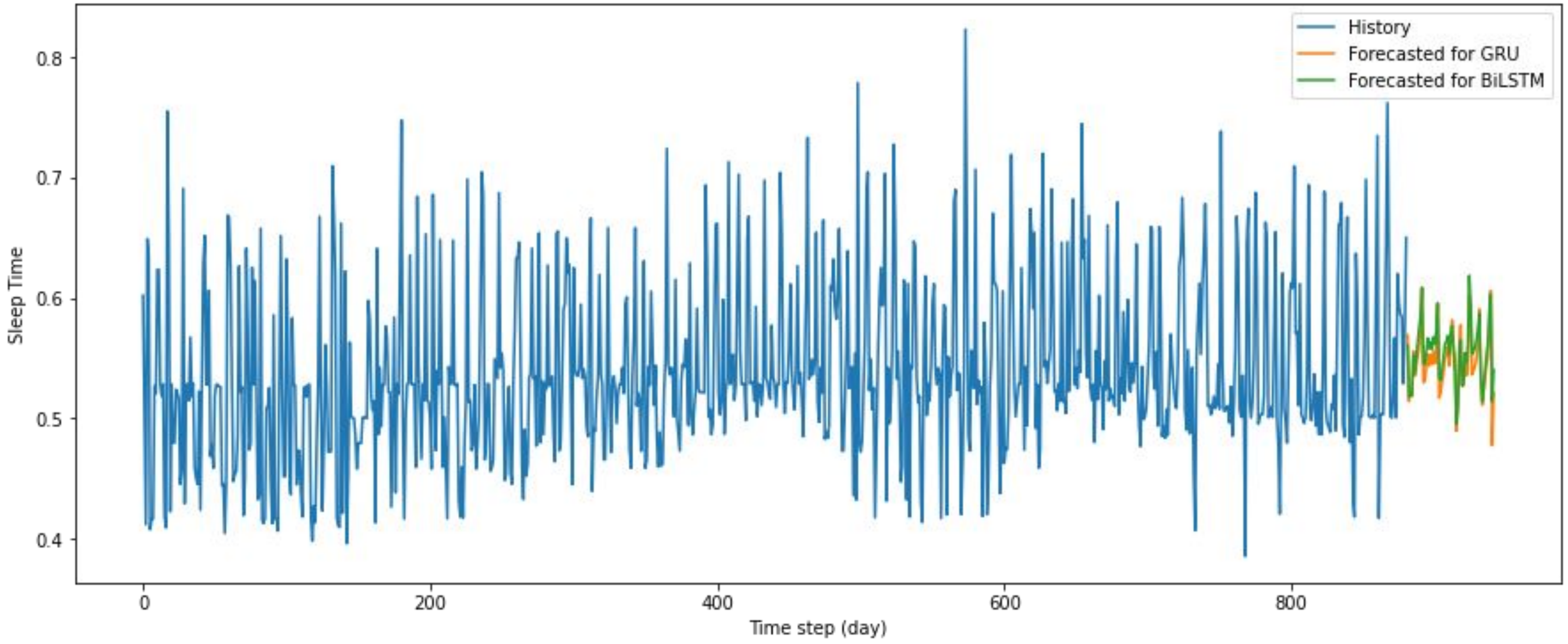
Sample Output - Sleep prediction

Actual vs predicted values



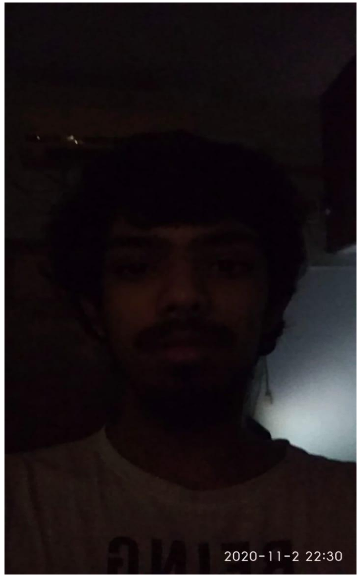
Sample Output - Sleep prediction

New predictions

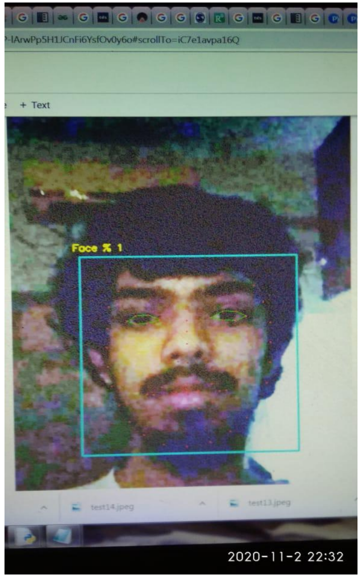


Sample Output - Luminance boosting

Low luminance



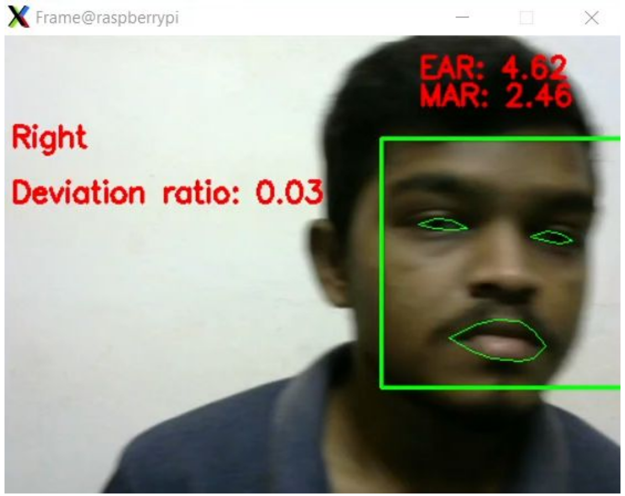
Increased luminance



Sample Output - Out of frame inspection

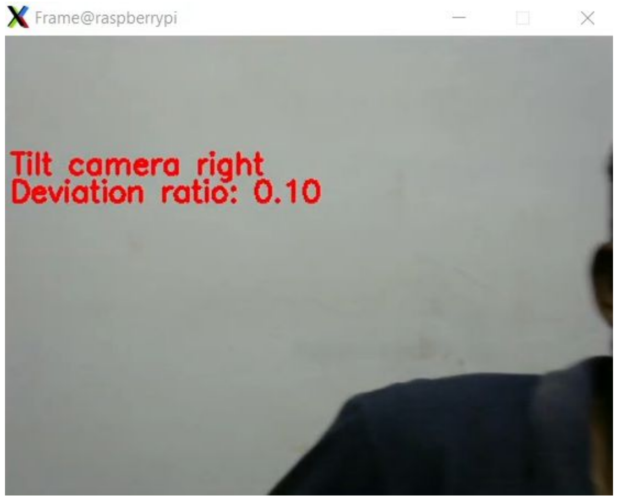
Last recorded position - right

Frame 1



Camera tilt suggestion - right

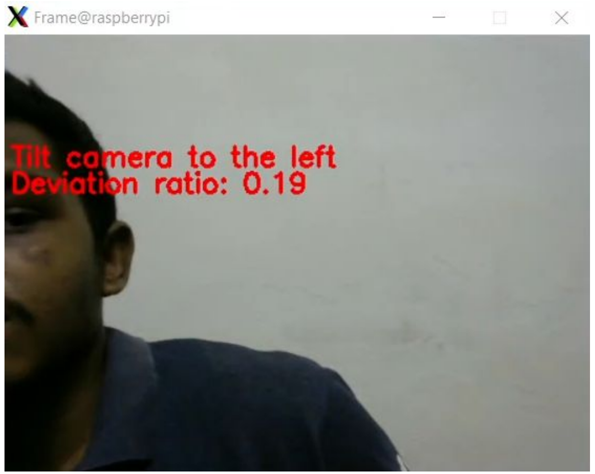
Frame 2



Sample Output - Out of frame inspection

Last recorded position - left

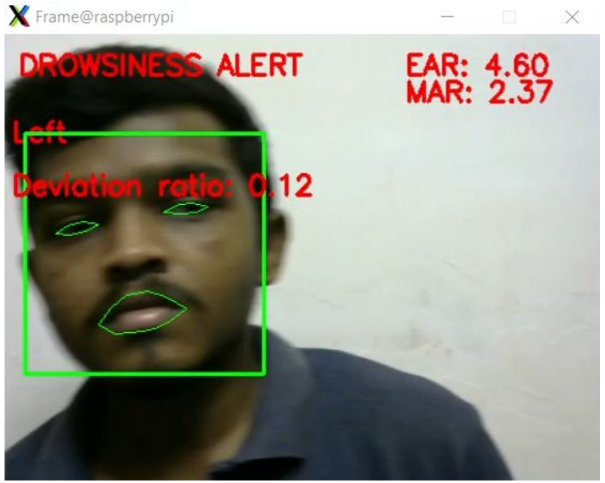
Frame 1



```
Tilt camera to the left
Deviation ratio: 0.19
```

Camera tilt suggestion - left

Frame 2



```
DROWSINESS ALERT          EAR: 4.60
                          MAR: 2.37
Left
Deviation ratio: 0.12
```

Conclusion

- The following modules have been successfully implemented in Raspberry Pi (v4, 2GB RAM):
  - Drowsiness detection
  - Drowsiness prediction
  - Luminance boosting
  - Out of frame inspection
- This system can detect drowsiness in real time and also easy to set up and relatively budget friendly.
- As future work, more powerful hardware can be used which can run more complex models with higher accuracies at higher frames per second and this system can be ported to run on smartphones to compare performances.

References

[1] Wanghua Deng , Ruoxue Wu, "**Real-Time Driver-Drowsiness Detection System Using Facial Features,**" 21 August 2019 doi: 10.1109/ACCESS.2019.2936663

[2] W. Walter, "**Overview of research on driver drowsiness definition and driver drowsiness detection**", Proc. Int. Tech. Conf. Enhanced Saf. Vehicles, pp. 462-468, 1995.

[3] M. Omidyeganeh, A. Javadtalab and S. Shirmohammadi, "**Intelligent driver drowsiness detection through fusion of yawning and eye closure**", Proc. IEEE Int. Conf. Virtual Environ. Hum.-Comput. Interfaces Meas. Syst., pp. 1-6, Sep. 2011.

[4] A. Dasgupta, D. Rahman and A. Routray, "**A smartphone-based drowsiness detection and warning system for automotive drivers**", IEEE Trans. Intell. Transp. Syst..

[5] Shuang Wang , Guanyu Wen , Hua Cai, "**Feature extraction and face recognition algorithm**," 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), doi: 10.1109/FSKD.2017.8393059

[6] R. Grace, V. E. Byrne, D. M. Bierman, J.-M. Legrand, D. Gricourt, B. K. Davi, et al., "**A drowsy driver detection system for Heavy vehicles**", Proc. 17th AIAA/IEEE/SAE. Digit. Avionics Syst. Conf. (DASC), vol. 2, pp. I36-1-I36-8, Oct. 1998.

[7] B. Alshaqaqi, A. S. Baquhaizel, M. E. A. Ouis, M. Bouumehed, A. Ouamri and M. Keche, "**Driver Drowsiness Detection System**", IEEE International Workshop on Systems Signal Processing and their Applications, 2013.

[8] A. Abas, J. Mellor and X. Chen, "**Non-intrusive drowsiness detection by employing Support Vector Machine**", 2014 20th International Conference on Automation and Computing (ICAC), pp. 188-193, 2014.

[9] N. Dalal and B. Triggs, "**Histograms of Oriented Gradients for Human Detection**", IEEE conf. on CVPR, 2005.

[10]M. Eriksson and N. Papanikolopoulos, "**Eye-Tracking for Detection of Driver Fatigue**", Proceedings of the International Conference on Intelligent Transportation Systems, pp. 314-319, November 1997.