Drowsy Truck Driver Detection

Rhea Anand
PES University
Bangalore 560085
Email: rheaanand96@gmail.com

N L Ramya
PES University
Bangalore 560085
Email: nlramya96@gmail.com

Mohammed Irfan
PES University
Bangalore 560085
Email: nlramya96@gmail.com

***Abstract*—This paper focuses on methods to detect if a driver is drowsy. It assumes that a camera is monitoring the driver as he is driving. The video is then fed to our system to extract frames, perform processing and detect drowsiness.**

We take two different approaches to detect the above. In the first approach we use the drivers eye as a feature. Detection of blinks help us determine if the driver is drowsy with certain thresholds set. The second approach is to detect the driver's mouth and track his/her yawns.

We use algorithms to set threshold and repeat the above steps to set off an alarm indicating that the driver is sleepy.

I. Introduction

A. Motivation

Recently, driver drowsiness has been one of the highest reasons for road accidents. Around 20 percent of all road accidents are fatigue related and 50 percent on some roads.

Driver drowsiness detection aims at preventing these accidents caused by the driver getting drowsy by identifying so and setting off an alarm to warn the driver. There are many ways to detect drowsiness of the driver:

- Vehicle Based : Steering patterns ,Lane monitoring
- Behavioural Based : Driver eye/face monitoring
- Physiological Based : ECG, EMG etc

B. Aspects we are focussing on

This paper aims at studying only the Behavioural Based approach. Statistics show that when a driver is drowsy, his/her

- Eyelids droop and head starts to nod.
- Yawning becomes almost constant and vision seems blurry.
- Blinks are hard and long

Our system studies all of these for various test cases.

C. Basic Workflow

The complete high level view of the Work flow with all the steps we took is shown in Fig.1.

The work flow starts with reading the video and extracting each frame of the video.

We use the Viola Jones face detection algorithm. The algorithm does the following:

Using haar cascading features we are able to detect the features of a person (i.e the face nose mouth etc) There are 5 types of haar kernels that are used to search for basic face features. These are convoluted over an image window of the persons face for the detection of these features. The kernels are increased in size gradually to account for different sized images. the combinations of these kernels and their different dimensional possibilities are 160,000. To decrease the redundancy of their predictions the algorithm uses auto-boosting that drops unnecessary kernels. A group of these kernels make weak classifiers. And a combination of such weak classifiers make a strong classifier that gives us a valid detection. The image window is passed through each weak classifier. At any point if there is a mismatch the algorithm doesn't proceed further to detect a face in the window.

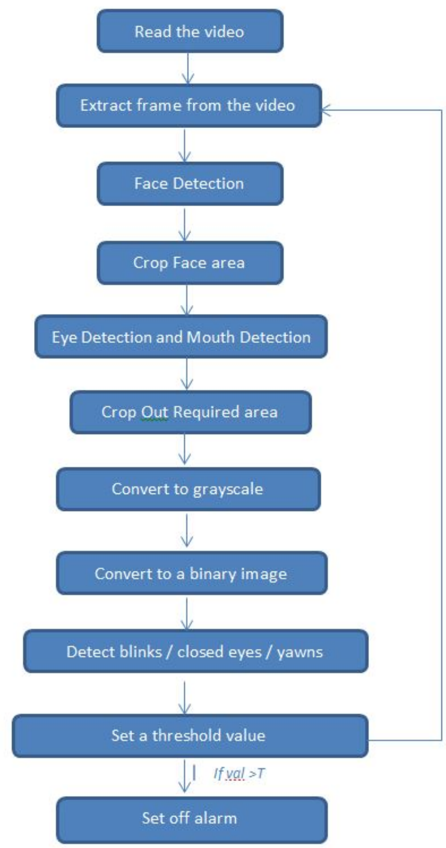

Fig. 1. Workflow

We use inbuilt Matlab functions to detect the face area and

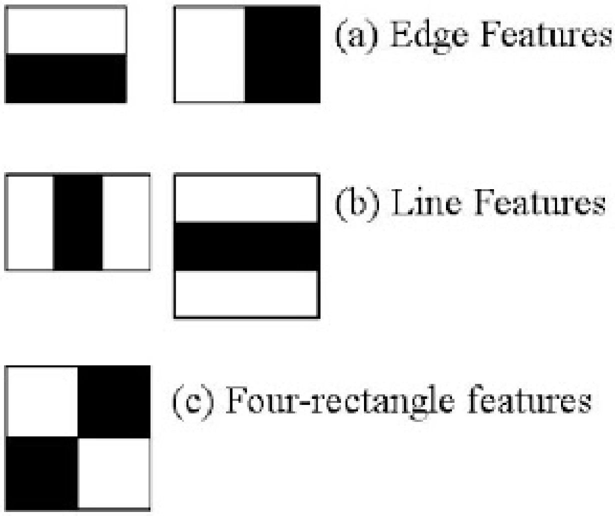

Fig. 2. Haar Features

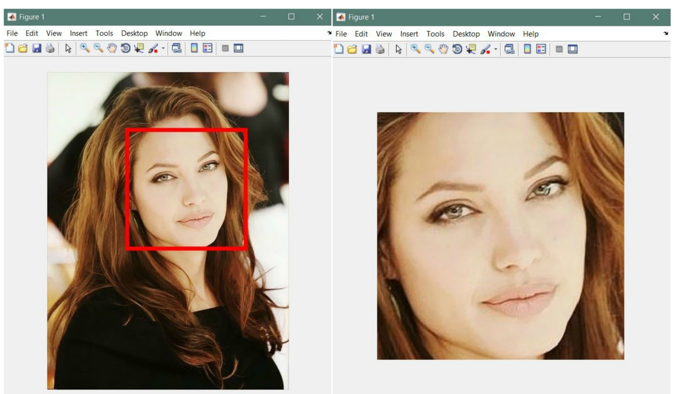

Fig. 3. Face Detection

Fig. 4. Eye Detection

crop it out to avoid any background interruptions on the further steps.

The next step is to detect the Eye. We use the same algorithm as we did for face recognition. We crop out this region to get images with just the eye. This is further elaborated in section II.

We apply the same steps for mouth detection to get just the mouth region. This is further elaborated in Section III

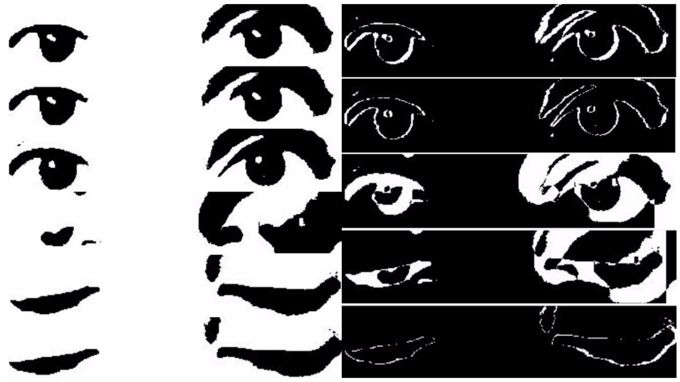

Fig. 5. Actual and Difference images

II. Detection of Blink

We convert the cropped eye image to black and white. The next step in the work flow is to classify each image as open eye or closed eye.

- Finding the differences between consecutive frames
- Training a classifier

To conclude that the driver is drowsy, we use the following approaches:

- An average person blinks about 15 times per minute. We count the number of blinks every few seconds and compare it to this. If the frequency of blinks is unusual, we conclude that the driver is drowsy.
- Every time there's a blink, we keep a timer for how long the drivers eye has been closed. If the time if beyond a threshold, we conclude that the driver is drowsy.Training a classifier

1) Finding Difference between consecutive frames: We take every two consecutive frames and find the difference between the two. We can't see many white pixels in the difference image unless there is a noticeable change. A noticeable change would either mean that the person has blinked or that the person has opened his eye after a blink. We set a threshold for the number of white pixels we require. If the number of white pixels is greater than the threshold, we increment the number of times there's been a change.

2) Training a Classifier: After extracting each frame from a video, we manually classify each image as open or closed eye. We then separate our dataset of images into training and test data - with 30 percent being training data and the rest 60 percent being test data. We do this split randomly. This makes our training and test data different each time we run the program. Using SVM Classifier, we generate a model based on a bag of features. We find the accuracy of the model using confusion matrices. The next step is to feed the video and to get labels for each new frame. We keep track of how long an eye has been closed continuously by resetting a count variable

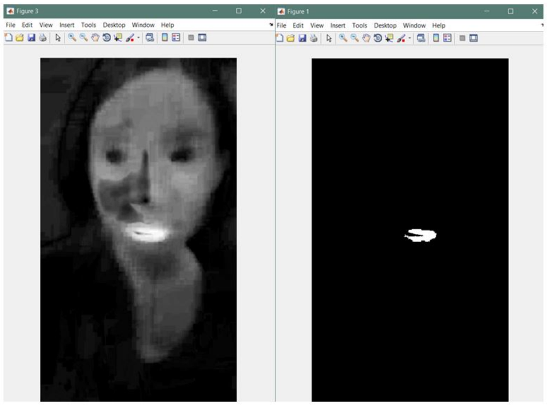

Fig. 6. Actual and Difference images

each time the eye opens. If the count is beyond a certain threshold value, we conclude that the driver is drowsy.

III. Detection of Yawn

Before we detect yawns, we need to detect the mouth region. To detect the mouth, we use 2 approaches:

- Use the viola Jones algorithm as we did for the face and eye. This method can be used to detect the mouth but fails to detect yawns.
- Find the mouth area based on the colour of the lip. Segment out the red area as the mouth. extract the red plane and subtract it with the gray scale/blue plane/green plane image to get high intensity values only around the lip region. This can then be labelled as the lip. Figure 6 shows how this works. The red Channel is subtracted with the green Channel in this case to localise the lip. This method doesn't work unless there is such a drastic change.

The other method we've used to detect yawns is to train a classifier: This is similar to training a classifier to classify blink. We use the same SVM classifier with a different dataset.

Once the yawn is detected, we find the frequency of the yawn. If it crosses a threshold, then we set off the alarm.

IV. Experiments and Evaluations

1) Blink Model: We test the blink model and eye model with multiple test cases. With respect to the blink model, we test our model with images of:

- Different eye colours
- Different sizes

The first method of finding differences works well with all the above cases as differences are noticed irrespective of the above conditions. The threshold values for the number of white pixels have to be changed for different types.

In the second method of training a classifier, we initially have a test data set of only images of a particular eye colour and size. This model gives us great results for different test cases of similar size and colour. The confusion matrix for this along with average accuracy for each run is displayed in Fig.5. Each run picks a random set of training and test data as mentioned previously. This reduces bias and over fitting. Taking the average of the runs. we get an accuracy of about 0.92.

This works surprisingly well with images of different eye colours as well. 36/40 frames were classified correctly in the first run and 35/40 in the second run.

When we feed in images of smaller eyes, 13 out of 30 were detected correctly in the first run and 16 out of 30 in the second run. This gives us a terrible accuracy and the model isn't very reliable.

After adding images of smaller eyes to our dataset, we get better results as displayed in Fig 6. Two runs are displayed in the image. With multiple runs, we get an average accuracy of 0.82.

On adding images of different colours to the dataset too, we get an average accuracy of 0.80.

PREDICTED

KNOWN	Closed	Open
Closed	0.86	0.14
Open	0.05	0.95

* Average Accuracy is 0.91.

PREDICTED

KNOWN	Closed	Open
Closed	0.91	0.09
Open	0.05	0.95

* Average Accuracy is 0.93.

Fig. 7. Results with only one type of image in dataset

PREDICTED

KNOWN	Closed	Open
Closed	0.94	0.06
Open	0.22	0.78

PREDICTED

KNOWN	Closed	Open
Closed	0.97	0.03
Open	0.42	0.58

Fig. 8. Results with images of smaller eyes added to the dataset

2) Yawn Model: We face certain problems with the first yawn method. Usage of the Viola Jones algorithm fails to detect yawns. It detects closed mouths perfectly for all of our test cases - Different sizes and colours do not affect this method.

Classification based on colour fails terribly for the test cases where we do not have a stark difference in colour between the skin and the mouth.

PREDICTED

KNOWN | Closed Open

Closed | 0.98 0.02

Open | 0.37 0.63

* Average Accuracy is 0.81.

PREDICTED

KNOWN | Closed Open

Closed | 0.95 0.05

Open | 0.36 0.64

* Average Accuracy is 0.79.

Fig. 9. Results with images of all different types added to the dataset

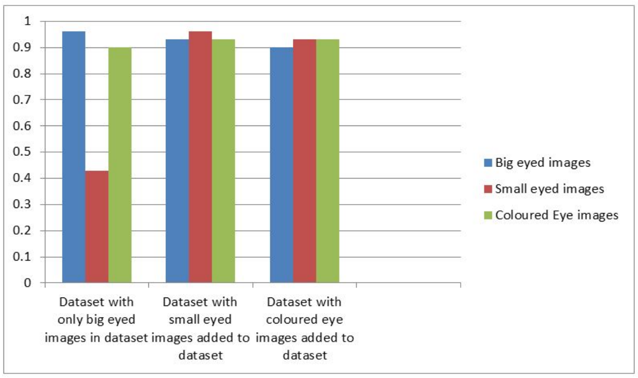

Fig. 10. Results with images of all different types added to the dataset

Using the classifier with a small dataset gives us a high accuracy of 98 percent. On testing this model with images of different types - different lighting conditions different eye sizes, we accuracy drops drastically. Even on including different types of images in the training set, the accuracy is still about 80

V. Observation and Discussions

Figure 9 shows the ratio of number of correctly classified images to total number of images for a test set of 30 images. This is displayed in the form of a column chart for different training data sets.

On adding new coloured eye images, the accuracy of the model actually drops. This is because the classifier gets confused with more of a variety of pictures. There's more of a conflict between images in the open set and images in the closed set. This reduces the accuracy.

The algorithm to detect the lip based on colour fails as there isn't much of a gradient colour difference.

The classification of yawn using SVM classifier isn't very accurate. With respect to the blink model,the model would perform better if the mouth area was already localised and the classifier worked on the cropped image. This would reduce dependencies from the rest of the image.

VI. Conclusion and Future Enhancements

1) Conclusions:

The project has presented various methods for drowsy driver detection and has compared the various techniques. The method of finding differences between frames gave us the best results for detecting a change. It works well for all test cases. Although, the classification model measures how long the drivers eye has been closed giving us better results in reality.

With respect to the various yawn models that we have employed, the classifier works best.

2) Future Enhancements:

- Incorporation of the drivers head drops / unusual body movements.
- Different colour of lips Better Yawn detection
- Take into account steering patterns and lane monitoring.

References

[1] *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.458.7214&rep=rep1&type=pdf*

[2] *https://ntl.bts.gov/lib/jpodocs/repts_te/7068.pdf*

[3] *https://pdfs.semanticscholar.org/71bc/acba6bbd44ef330432ce1603c8874ca35d03.pdf*

[4] *https://www.ri.cmu.edu/pub_files/pub2/grace_richard_2001_1/grace_richard_2001_1.pdf*

[5] *https://www.youtube.com/watch?v=_QZLbR67fUU*

Contribution of each member

Team Member	Contribution
Mohammed Irfan	• Face and Eye detection • Documentation • Code execution • Presentation
N L Ramya	• Blink and Yawn Detection algorithms • Experimentation and Evaluation • Visualisation • Report writing
Rhea Anand	• Blink and yawn Detection algorithms • Documentation • Code execution • Blog updation

Acknowledgment

We would like to thank Ms. Gowri Srinivasa for this opportunity and for providing us with her valuable feedback and unlimited support throughout the time line of the project.