Driver Alertness Detection System

Jai kumawat
E18CSE072
jk1289@bennett.edu.in
BTech, Computer Science and Engineering
Bennett University
Greater Noida, India

Digvijay Singh
E18CSE049
ds5864@bennett.edu.in
BTech, Computer Science and Engineering
Bennett University
Greater Noida, India

Shaleen Agarwal
E18CSE165
sa4531@bennett.edu.in
BTech, Computer Science and Engineering
Bennett University
Greater Noida, India

***Abstract*— A sound working real time driver's alertness system requires a new intelligent algorithm and parameters in order to detect drowsiness and fatigue more accurately. Previous models just considered facial features for detecting drowsiness, but now we are considering heart rate for a clearer indication of fatigue or lack of alertness. In this paper we will be using a unidirectional Long Short-Term Memory (LSTM) architecture for implementation of drowsiness detection through facial features. Since the data is sequential having long range dependencies the LSTM model will be useful in this case. And for heart rate a plethora of algorithms were used for face tracking, region of interest extraction, signal extraction, detrending and processing, filtering, etc. which will be discussed thoroughly in the paper. We discuss our implemented models and conclude our discussion with interesting results. Alongside modelling, a comprehensive data analysis on the features of the dataset itself using our understanding of correlation, variance and similar such factors for a wider picture is done for better modelling. Furthermore, our limitations during the implementations are discussed and our future plans to elaborate our dataset for robustness of the model.**

Keywords: Drowsiness Detection, LSTM, MAR, MOE, Heart-Rate Detection, JADE, ICA, FFT

I. INTRODUCTION

In this age where population is increasing rapidly the road mishaps are also increasing rapidly. One of the foremost reasons being lack of driver alertness, this is caused due to driver's fatigue which alone accounts for 25 percent of motor accidents and around 60 percent of these accidents causes death or serious injury. A main reason for this lack of alertness or fatigue is sleeplessness, insomnia, drinking and driving, etc. Therefore, the driver's fatigue and drowsiness state play a key role in road mishaps that take thousands of lives annually.

In recent years, intelligent algorithms are being developed for the upcoming cars using new systems and features. These systems use WSNs to monitor and transmit the condition of the car and the driver. Smart cars that use software techniques to control car dynamics like engine speed, steering, transmission, brake etc. have improved the quality and essence of driving significantly. Ad hoc networks were the first to develop the automatic navigation systems in cars. A noticeable weakness of these systems is that they do not respond to the change in the surroundings and environment in real time. It is especially important in driving where time is a critical factor in a driver's decision. On the other hand, another method to check the driver fatigue is monitoring the physical condition and facial expressions of the drivers, which wireless sensor networks are unable to process and transmit this information with ample precision and have an adequate recall. Hence it is very important and crucial to develop an efficient driver's alertness system.

Monitoring driver's alertness is a tricky problem as we need to set up parameters for when to consider someone fit for driving. For this the paper proposes a model that combines drowsiness detection through facial features and heart rate detection through camera and provides a comparative real time study of heart rate and drowsiness. Heartbeat has several properties from which you can tell a lot about a person by reading his heartbeat. For example, Steady constant heartbeat during strenuous activity is linked to a good healthy heart, Low resting pulse can be a good indicator of a healthy cardiovascular system and much more. Nowadays there are a lot of cheap and effective methods to monitor heart rate. As most Smart bands come equipped with a sensor. Most mobiles, laptops and even some cars are shipped with inbuilt front facing cameras these days. This can also be used to determine an accurate measure of heart rate. This is done by skin extraction and signal filtering with FFP and ICA . So, in this project we will be using Photoplethysmography and direct approaches to measure heart rate and comparing their result. We will combine heart rate with a drowsiness model for final output

.

The rest of the paper shall be organized as follows:Several approaches have been proposed to accurately judge a driver's ability to drive the car. These approaches can be broadly classified into two groups: Vehicle-based approaches, and Visual feature-based approaches which will be discussed in Section II.In Section III,we will look more into the details surrounding the dataset used for implementing the models in terms of its construction, features, labels, and in a separate section the pre-processing done before modelling. In Section IV,we shall investigate the models being used and their specific methodologies with respect to the understanding of the dataset as laid out in

Section III. In Section V, we will explore the results we have found, and then Section VI where we discuss our limitations.Section VII, we shall propose our future prospects and goals in order to create a more robust model. And finally, we conclude our discussions with the results we found in Section VIII.

II. RELATED WORK

Several approaches have been proposed to accurately judge a driver's ability to drive the car. These approaches can be broadly classified into two groups: Vehicle-based approaches, and Visual feature-based approaches.

A. Vehicle-based approaches:

These approaches use sensors fixed on the vehicle itself. They keep track of any abnormal patterns in driving such as unsteady pedal pressure, erratic steering wheel movements, weaving car, high speed etc. Using these patterns and leveraging ML (Machine Learning) techniques, they try to predict the probability of an accident happening.

Yu et al [1] uses smartphone sensors such as the accelerometer and the magnetometer to monitor erratic driver behavior , using the features namely acceleration and orientation extracted they combine a SVM (Support Vector Machine) and a NN (Neural Network) to build a classifier which conducts fine-grained abnormal driving behavior detection. Samiee et al [2] combines vehicle lateral position, steering wheel angle and eye tracking techniques to build a drowsiness detection model which can function even if any of the methods fail making it particularly robust to input signal loss. While [3] uses a combination GMMs (Gaussian Mixture Models) and HMMs (Hidden Markov Models) leveraging sensor data such as pedal pressure, pedal position as well as the car's movement behavior using GPS to model driver behavior.

B. Visual feature-based Approaches:

Visual feature-based approaches take note of visual information inferred from the driver's face, eyes, body, gestures etc. Zhao et al [4] instead of just using singular features such as landmark points around the eyes or nose and training shallow classifiers instead proposes to use a DBN (Deep Belief Network) to address the problem of generalization which the above mentioned shallow classifiers suffer from. First the landmarks and textures of the facial region are extracted then a DBN is trained using them which classifies the images on the basis of facial drowsiness expressions [4]. Verma er al [5] also uses a similar approach as [4] but instead using two VGG16s to extract facial as well as geometric features to classify a driver's emotional state. Mandal et al [6] uses a multitude of features consisting of head-shoulder detection, face detection, eye detection, eye openness, drowsiness measure percentage of eye closure (PERCLOS) estimation to measure a bus-driver's fatigue. While Xing et al [7] uses a FFNN (Feed Forward Neural Network) in conjunction with a Kinect camera to identify a driver's postures classifying them as either safe or unsafe.

III. DESCRIPTIVE ANALYSIS OF THE DATASET

A. Dataset Description

The driver facial landmark-feature classifier is trained on the UTA-RLDD (The University of Texas at Arlington Real-Life Drowsiness Dataset) dataset [8]. The RLDD dataset's main purpose is for the task of multi-stage drowsiness detection. It targets not only late or early stages of drowsiness but also the subtle-cases in-between. This can be helpful in early detection of drowsiness for the drowsiness preventative measures to activate [8]. These subtle expressions are associated with instinctual and physiological changes which can be hard to accurately capture by an actor. The RLDD dataset attempts to address this problem by tracking the activities of real-life subjects.
The RLDD dataset is made up of 30 hours of RGB videos of 60 healthy participants over a wide range of ethnicities and age. For each participant one video each belonging to 3 different states was obtained. The classes are alertness, low vigilance, and drowsiness which are further classified into different labels based on the magnitude of the class. The videos were recorded by the participants from different environments and backgrounds using a phone or a web camera keeping the fps at or below 30 to simulate the performance of normal cameras used en masse.

State Description:-

1. Alert: In the dataset, alert state is classified as when the subject was completely conscious and could drive for long hours.
2. Low Vigilant: this state corresponds to subtle cases when some signs of sleepiness appear, or the subject is feeling sleepy but has no problem keeping alert. Driving is discouraged in this state even if it is possible.
3. Drowsy: This state implies that the subject must actively try to not fall asleep.

These states are further divided into 9 different labels [8].

Table 1: Labels associated with the States

Label	State
1	Extremely Alert
2	Very Alert
3	Alert
4	Rather Alert
5	Neither Alert or Sleepy

6	Some signs of sleepiness
7	Sleepy, but no difficulty remaining awake
8	Sleepy, some effort to keep alert
9	Extremely Sleepy, fighting sleep

The dataset consists of 180 RGB videos. Each video is around 10 minutes long and is labelled as belonging to three different classes: alert (0), low vigilant (5) and drowsy (10). The subjects labelled the dataset themselves according to the state they felt they were in. This simulates the transition from being alert to showing some signs of sleepiness and finally to drowsiness. The dataset is further divided into 6 folds of 12 participants each, for the purpose of cross-validation.

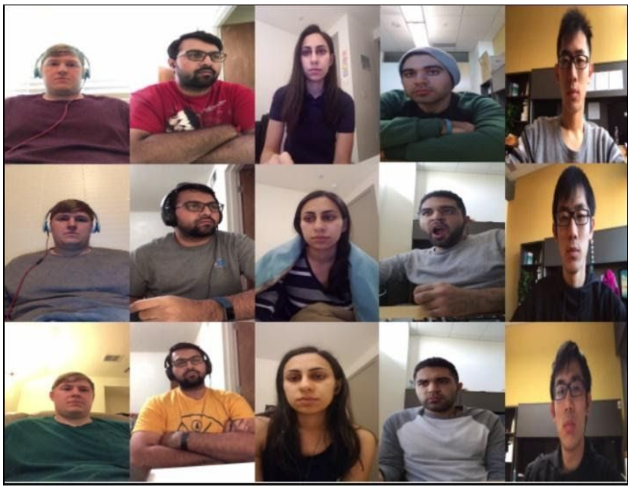


Figure 1: Sample images from dataset taken from the UTA-RLD website.

Fig 1. Consists of frames extracted from the video with the first row corresponding to alert state (0), second to low vigilant (5) and the third row corresponding to the drowsy (10) state.

B. Dataset Pre-processing

This project utilizes OpenCV to process the videos from the dataset. Frames are extracted from the videos starting from the 3-minute mark till the end to get an accurate reading of the subject's state. A frame is extracted every second from the video feed and stored in an array. Every video was around 10 minutes in length so around 300 frames were extracted from each video.

After the frames are extracted, dlib's [9] facial landmark detector function is called for each frame. The landmark detector is trained on the iBUG 300-W landmark dataset [10] using the implementation from [11] [12] [13].

The dataset consists of labelled images of faces annotated with 68 landmarks.

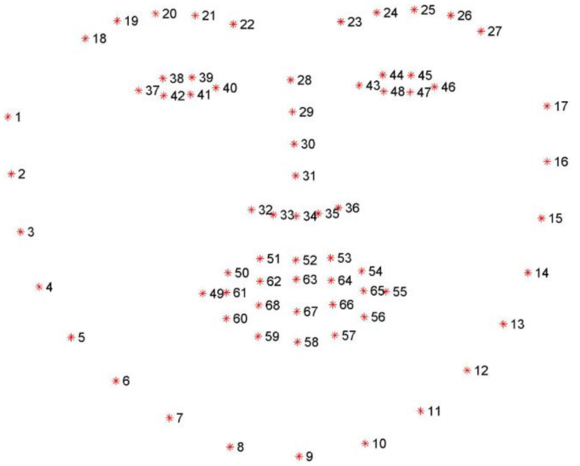


Figure 2: 68 landmarks annotation of a face

Of these landmarks extracted from the video frames, only the points 37-68 were i.e. the landmarks around the eyes and mouth were retained.

Feature Extraction

Based on the landmarks extracted earlier, several features were judged on their relevance and suitability for the model and in the end 4 features were concluded on [14].

1. Eye Aspect Ratio (EAR) :-

   As the name suggests EAR is the ratio of the eye's length to its width. This feature is particularly useful as the EAR when the eyes are open is approximately constant and rapidly falls to zero when the eye blinks. This gave an estimation of the frequency of the eye blinks allowing us to estimate whether a person is drowsy or not as if the person is alert, the frequency of the eye blinks would be less as well as the successive time interval between the blinks would be longer whereas the converse would be true if the person shows signs of drowsiness i.e. rapid eye blinks, eyes staying shut for a longer period of time.

   The EAR ratio is calculated as

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||}$$

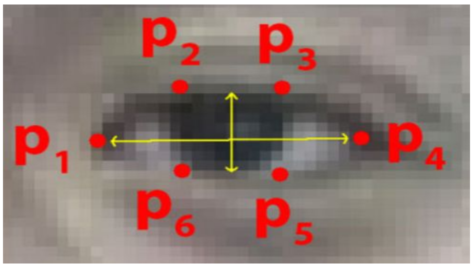


Figure 3: The 6 landmarks associated with the eye

2. Mouth Aspect Ratio (MAR):-

   Like the EAR, the MAR measures the ratio of the length of the mouth to its width. This feature is of significance in the way that if an individual becomes drowsy, they are more likely to yawn making their MAR to be higher, this ratio will also help us quickly identify if an individual is yawning in successive frames, helping us also to track the count of the frequency.

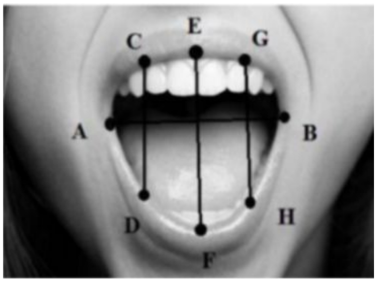


Figure 4: Points associated with the mouth

MAR can be calculated as

$$MAR = \frac{|EF|}{|AB|}$$

3. Pupil Circularity (PUC)

   PUC is a measure similar to the EAR, but it places greater emphasis on the pupil instead of the entire eye. If someone is feeling drowsy, their eyes will probably be half closed or not fully open, in that case the PUC will be smaller. The PUC can be calculated as

$$Circularity = \frac{4 \times \pi \times Area}{perimeter^2}$$

$$Area = \left(\frac{Distance(p_2, p_5)}{2}\right)^2 \times \pi$$

Perimeter = Distance(p1, p2) + Distance(p2, p3) + Distance(p3, p4)+ Distance(p4; p5) + Distance(p5; p6) + Distance(p6, p1)

4. Mouth aspect ratio over Eye aspect ratio (MOE)

   MOE is the ratio of MAR to the EAR. It will help us detect the more subtle changes in the MAR and the EAR because as an individual becomes drowsy, the MAR will increase as the EAR will decrease, this will exaggerate the changes taking place as the numerator decreases and the denominator decreases. Since we take the MAR as the numerator, so the MOE is supposed to increase as the drowsiness levels increase. MOE can be calculated as

$$MOE = \frac{MAR}{EAR}$$

Feature Normalization

When the model was trained on the above selected features, it gave a very low accuracy on the validation dataset implying it was not able to generalize well on the persons whose faces the model has not seen before. The features that we had extracted were specific to the individual as every person has their own unique facial characteristics i.e. Person A might have bigger eyes than Person B making the EAR of person A to be bigger than B's. If the model is trained using Person B, when tested on Person A it will predict A to be alert even if that's not case as it will be looking for a smaller EAR which will not be the case with A. Also if the number of persons are further increased with each having their own characteristics the model will have trouble finding relationships between the features leading to even more accuracy loss.

So, to make the model more robust to previously unseen faces, Feature Normalization was performed. To normalize the features [6], the first three frames of each person's alert video were extracted and their mean and standard deviation was calculated and were used to normalize every feature individually for each person.

Mathematically, the normalized equation was

$$\textbf{Normalised Feature}_{n,m} = \frac{\textbf{Feature}_{n,m} - \mu_{n,m}}{\sigma_{n,m}}$$

where:
n is the feature
m is the person
$\mu_{n,m}$ and $\sigma_{n,m}$ are taken from the first 3 frames of the "Alert" state

IV. METHODOLOGY

A. FOR HEART RATE DETECTION

1.1 Setup

For this experiment we used a camera sensor from several sources in the initial phase (webcam and android) , however for validation and testing of the final version, the laptop's webcam was used . All videos for heart rate extraction were taken at 10 frames per second at a resolution of 640 x 480 . The format was AVI and the standard 24 bit RGB was used .

For evaluation of our algorithm was measured in two phases :

a) During development the Public Benchmark Dataset for Testing rPPG Algorithm Performance [15] was used which consists of a heart rate (BVP sensor) measure of 60 people of various age groups under varying level of luminosity conditions, along with their stabilized (fixed) face . The videos from the dataset were downscaled from 1280 x 720 to 640 x 480 and the framerate was capped at 15 fps

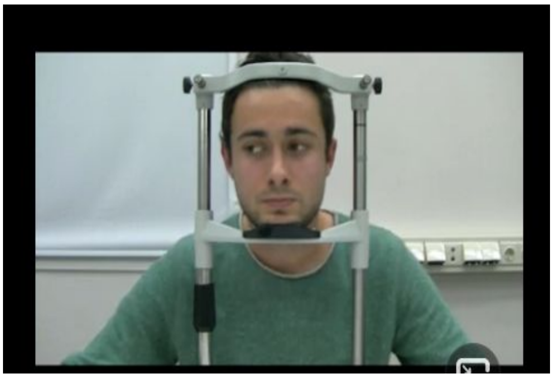


Figure 5: Dataset setup

b) During the final testing phase PPG sensor (Mi Band) was used with the laptop's webcam (480p ,15fps) was used . WE were seated in front of a laptop at a distance of around 0.5 m from the webcam.

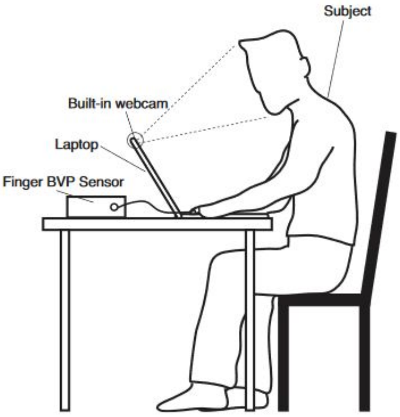


Figure 6: our setup

1.2 Algorithms used

Many algorithms were used to extract the heartbeat value from the face .

During the initial testing stabilized face was used (Fig 5) , however this cannot be expected to work in real-world use case scenarios, the face might move. Therefore, Open Computer Vision OpenCV 2.0 [16] was used to track face during motion .

In the video we have three colour components , we use the FFT(Fast Fourier Transform) method to calculate the average of R ,G ,B values from their respective channels in the video.

In the face-cut region we need to analyse the changes in face colour when blood pumps blood into the body , these changes will give out a light wave , the changes in this wave can be mapped to heart rate . However , the wave might also capture some other sources of light fluctuation . We use ICA(Independent component analysis) to fix this .

Due to the difference in haemoglobin absorptivity with visible spectral range , each channel will output a combination of the important signals with slightly different weights. Therefore we can assume that our video stream contains linear mixtures of the sources .

Let x(t) denote observed signals and s(t) denote underlying source signal then

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

. The aim of ICA is to find a separating x(t) value . Let W be the inverse of the original mixing matrix then

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$$

S contains the underlying source signal.

3 Feature Extraction

The main features are Red signals, Green signals and Blue signals which are produced from the red, blue and green colour values of pixels of Region of interest .

3.1 Reading Image

A video is a collection of frames . We need to convert the video stream to images to extract and analyze the data . For this we use a fixed video resolution and try to maintain a constant frame rate . For image extraction we use the novel keyframe video extraction algorithm which will split the video frame by frame .

3.2 Face Tracking

We use the python face tracking feature of the Open Computer Vision (OpenCV) library to obtain the face coordinates . This face detection algorithm is based on work by Viola and Jones ,Lienhart and Maydt . We use the pretrained front trained classifier which ships on default with the OpenCV 2.0. This model is called the Haar cascade model..

The cascade applies a set of simple classifiers to the area of interest sequentially. Using weighted vote it builds a classifier at each stage ,this method is known as boosting.

Since this is a real time detection model the coordinates of the area of interest are updated at regular intervals in order to identify the dimensions even when motion is present which changes the volume. When face is detected the algorithm returns a tuple (x,y,h,w) where x and y are the starting coordinates of the box and h and w are the height and width of the box , using this we can crop out the pixels we need .

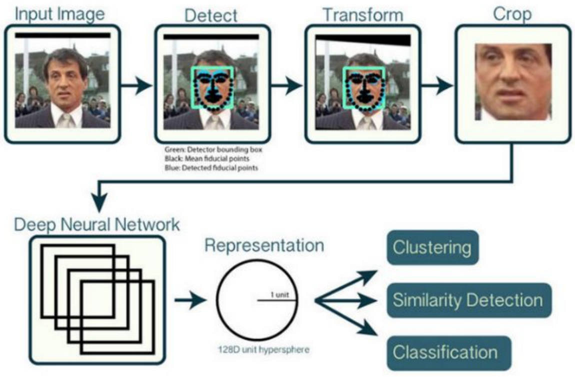


Figure 7: Representation of neural network

3.3 Region of Interest

We only want to process the RGB values from the area of the face , this is because other areas can add unwanted waves to our main wave which will further lead to inaccuracies due to impure data.

The OpenCV cascade discussed above returns an area which contains some impurities , in order to remove them we can use various methods.

The theoretically ideal algorithm would use the image segmentation algorithm. This will automatically cut out the face only region through image processing techniques. However this is very computationally expensive , a simple but lossy alternative would be to further crop the image around the centre to just extract face region cutting out some part of the face region.

This will lead to loss of some data but the algorithmic complexity and time complexity would both be integral. Therefore we use 60 % of the width and 80% of the height of the values returned by OpenCV .This will give us a region without impurity .

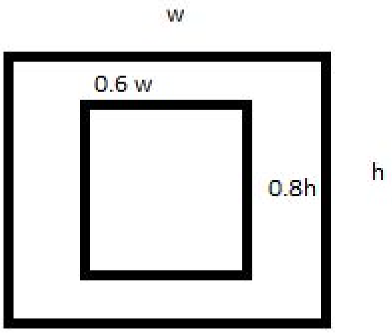


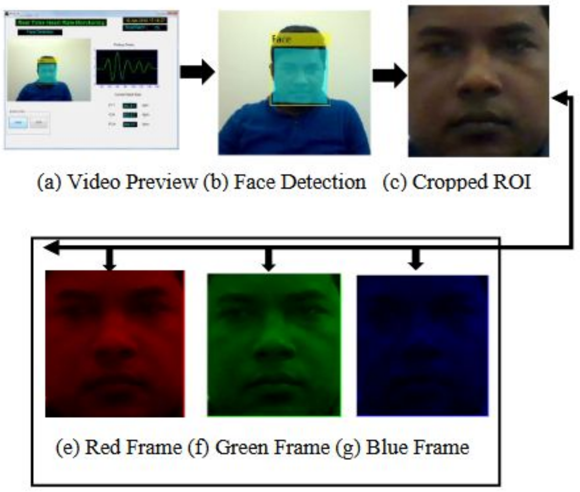
Figure 8

3.4 Signal Extraction

Each color pixel consists of 3 components, the Red, Green, and the Blue values, which is expressed as a 3x1 matrix where each column corresponds to the individual color's value.

We convert the images in two phases. First, we average out the values for each channel and in the second phase red, blue and green signals are calculated from the summation of these values .

These values can be seen in fig 9



(a) Video Preview (b) Face Detection (c) Cropped ROI

(e) Red Frame (f) Green Frame (g) Blue Frame

Figure 9: Work done so far

3.5 Signal Detrending

Some aspects of our data can cause changes in all the signals extracted over time . This might lead us to read some correlation in data which can cause the model to detect false relations and reduce the accuracy..

To avoid this we use signal detrending . In our case several environmental factors such as change in temperature, different skin colour or external noise can affect the data. Some hardware inconsistencies such as a change in camera sensor across various devices can also cause unexpected trends .We detrend the signal with the method discussed in [17] . This method is based on the **s**moothness prior's approach[21] with the smoothing parameter λ =10 and cutoff frequency = 0.059 Hz .

3.6 Filter

Before applying the wave algorithms we remove the temporary extreme signals which might be caused due to some unfavourable condition .
For this we use filtering . In our approach, signals are filtered by Hamming window (128 point, 1-2 Hz, for normal Heart Rate in range 50-120)

3.7 Normalization

For proper processing we need to normalize our data . We use the method discussed in [18]

$$X_i(t) = \frac{Y_i(t) - \mu_i(t)}{\delta_i}$$

Here u(t) is the mean and δ is the standard deviation

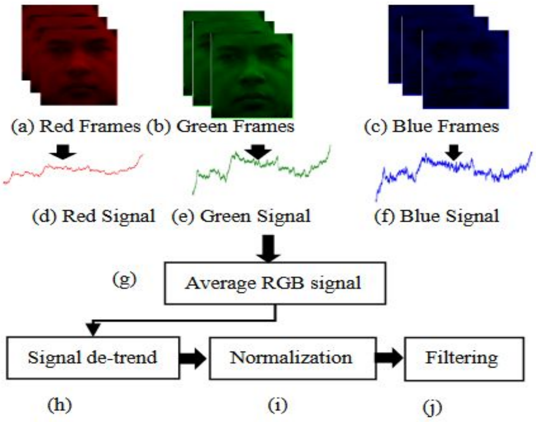


Figure 8: Signal Detrending

3.8 Real Time System

The parameters extraction methods discussed in [19] are used as a reference to extract the heart rate parameter in this paper . This method determined heart rate in a video of any given length .
However in order to use this with the other model , we need a real time model .
To do so we need to implement a buffer to store it incoming stream , it will act as a multilevel queue where latest data will be added and oldest will be removed in each iteration .
Through trial and error it was found that a buffer of length 50 performs best for our use case .
We create a buffer of length of 50 images and analyze it using our method in the same way . This will result in an output value based on the previous images and also introduce a delay . Since heart rate changes do not occur rapidly in seconds ,the delay can be ignored .
Hence the system will be able to extract images in real time

3.9 Signal Processing

The normalized raw traces will be decomposed into 3 signals using Independent component analysis [Fig 9].

For our research we use the **joint approximate diagonalization of eigen-matrices (JADE)** algorithm . Through various researches conducted independently, it is found that the Green signal gives the best result for the detection of blood volume .

Therefore , we use the green signal as the desired signal .

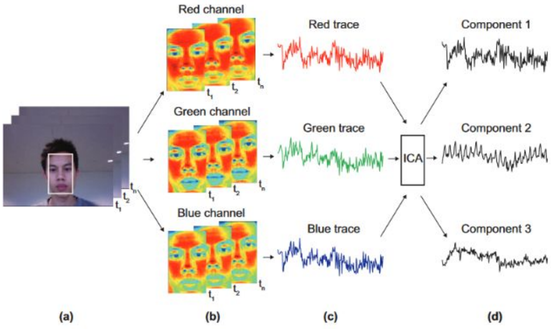


Figure 9: A ICA

Now we will apply the Fast Fourier transform (FFT) on the obtained signal to get the power spectrum. The heart rate frequency will now be mapped to the power spectrum. For our use, we set the operational range to 1-2 Hz, for normal Heart Rate in range 50-120 to provide a good range for heart rate measurement. This method can also be used to obtain blood pressure.

3.10 Heart Rate from heart rate frequency

The heart rate frequency amplitude and power is very inaccurate and is of no use alone .

However it does roughly resemble the heart rate signal wave from a graded medical device .

Thankfully, we are only concerned with the number of maximas in our time frame.

Therefore, the number of maximas in the given timeframe is equal to the number of heartbeats in that range

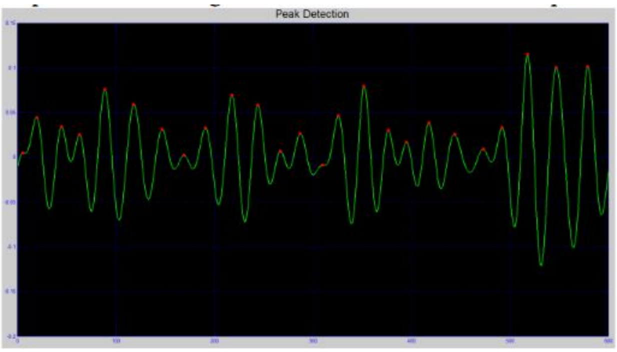


Figure 10: Maxima Counting

Finally we obtained heart rate by counting the number of peaks in a timeframe and scaling it to 60 seconds which will give us BPM(beats per minute)
For Fig[10]

$$\begin{aligned} \text{HR} &= 60^* f_h \text{ bpm} \\ &= [60 \text{ x (number of Peaks/Time)] bpm} \\ &= 60 \text{ x } (25/20) \text{ bpm} \\ &= 75 \text{ bpm.} \end{aligned}$$

We finally have a heart rate for the last 50 frames.

B. *FOR DROWSINESS DETECTION*

This project will be using a unidirectional Long Short-Term Memory (LSTM) architecture for implementation. Since the data is sequential having long range dependencies the LSTM model will be useful in this case.

Long Short-Term Memory (LSTM)

LSTMs are a special type of Recurrent Neural Networks (RNNs) which specialize in learning long-range dependencies, mainly used to process entire sequences of data instead of singular data points which traditional Neural

Networks are mainly known for. Recurrent Neural Networks have feedback connections possessing an internal memory which allows them to “remember” information from past sequences of data.

How can RNNs have an internal memory space?

The answer to this question is that when making a decision, a RNN not only considers the present input but also uses the previous input’s output to make the decision, this allows the RNN to “remember” the data as it’s taking that into account too when making the decision. This is also the unique feature of RNNs and other Neural Networks. In other Neural Networks, the data points are discrete and independent of each other but in a RNN they are continuous sequences of data.

Traditional RNNs suffer from the vanishing gradients problem when exploring long range dependencies. LSTMs were chosen for implementation because they are proven to be particularly effective for keeping track of long-range dependencies without worrying about the vanishing gradients problem.

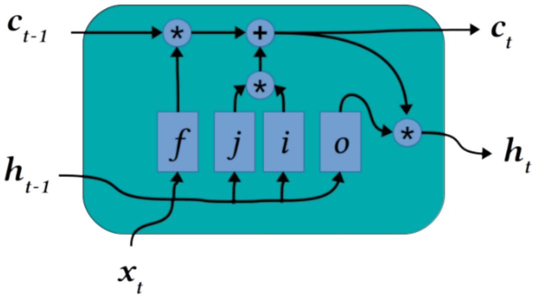


Figure 11: Structure of a LSTM network

The LSTM network consists of 3 “gates”. Each gate is responsible for how the information fed to the network is to be processed. The three gates of LSTM for each timestep are: Forget Gate, Input Gate, Update Gate.

1. Forget Gate: As the name suggests, this gate is responsible for deciding what part of “internal memory” of the network should be deleted, or in other words what part of the information should be retained.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

2. Input Gate: Input Gate decides what part of the present input should be taken into consideration during the current time step.

$$i_t = \sigma\,(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C} = tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. Output Gate: This gate combines the present input and the input from the previous states.

$$o_t = \sigma\,(W_o\,[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh\,(C_t)$$

Figure 4: Equations of Output Gate

Model Structure

First the input features are converted into batches of size 5 since a time-step of 5 is being used, in other words the LSTM will only take those data into consideration that are at most 5 timesteps behind.

After the batches are obtained, it is fed to the LSTM cell with 512 hidden units followed by a Flatten layer which converts the LSTM output to a 1-D array, Afterwards the output is fed to a couple of Fully Connected layers with Sigmoid activation before arriving at the Output Layer. In between the Fully Connected layers, Dropout Regularization is used to force the model not to depend on the location of the features but the size and shape of the features themselves in order to address the problem of generalization which was discussed earlier.

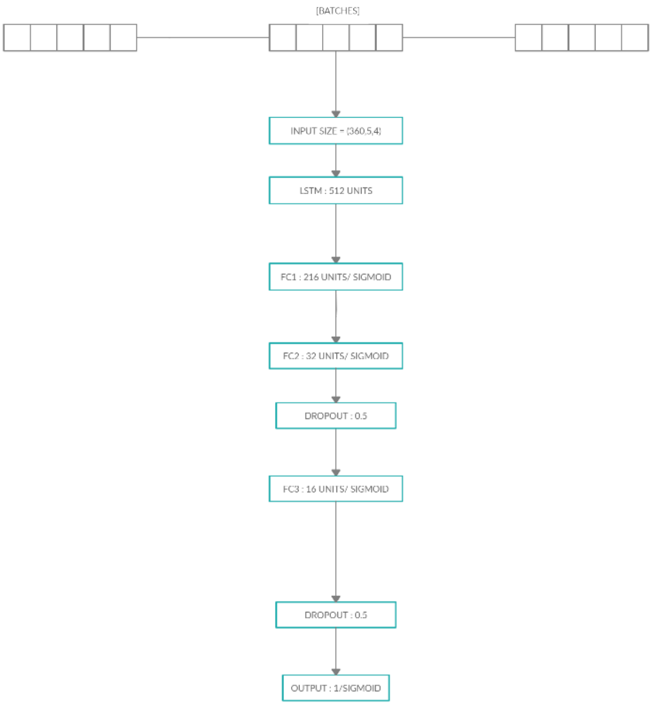
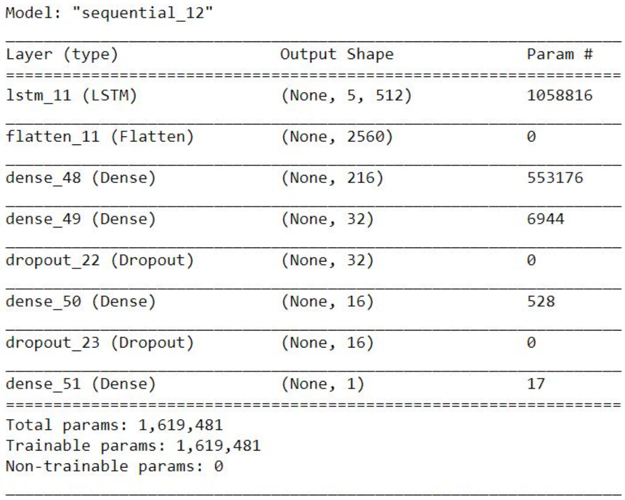


Figure 12: Model Work-Flow Structure

Table 2: Model Parameters

Number of Epochs	50
Learning Rate	0.00005
Timestep	5

Table 1: Model Parameters



```
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_11 (LSTM)               (None, 5, 512)            1058816   
_________________________________________________________________
flatten_11 (Flatten)         (None, 2560)              0         
_________________________________________________________________
dense_48 (Dense)             (None, 216)               553176    
_________________________________________________________________
dense_49 (Dense)             (None, 32)                6944      
_________________________________________________________________
dropout_22 (Dropout)         (None, 32)                0         
_________________________________________________________________
dense_50 (Dense)             (None, 16)                528       
_________________________________________________________________
dropout_23 (Dropout)         (None, 16)                0         
_________________________________________________________________
dense_51 (Dense)             (None, 1)                 17        
=================================================================
Total params: 1,619,481
Trainable params: 1,619,481
Non-trainable params: 0
_________________________________________________________________
```

Figure 13: Model Summary

C. *FINAL MODEL*

The final model implemented takes input images and outputs the classification values. However for our use case we need to obtain and output values in real-time .

We observe that the model takes a set of 5 images over a period of 5 seconds and outputs a value according to it.
We can output a singular integral value for every 5 seconds, however that will be very inefficient and can cause a delay up to 10 seconds for representing the changes.

To combat this we use a multi-level queue of size 5 which will act as a buffer where the latest value will be added and the oldest value will be removed for every second. This buffer will be analysed every second and the output value will be produced every second.

The delay for this method will be 1 second, which is a lot faster then our previous approach

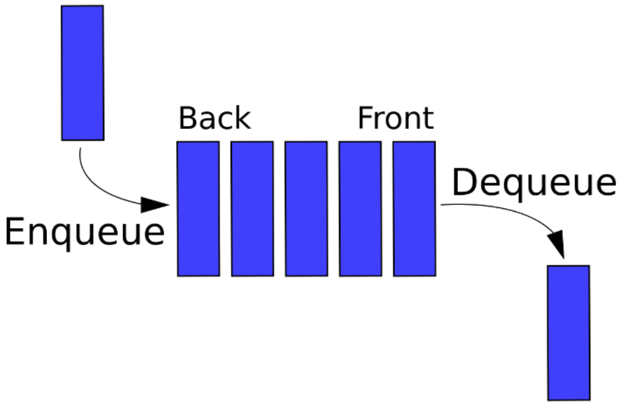


Figure 13: Method used

V. Results And Discussion

Performance of the Heart Rate Model

For testing we evaluated our algorithm using the Public Benchmark Dataset for Testing rPPG Algorithm Performance [15] which consists of heart rate(BVP sensor) and stabilized face of 60 humans of different age groups . We ran our algorithm on the video and compared the average heart rate for 12 participants.

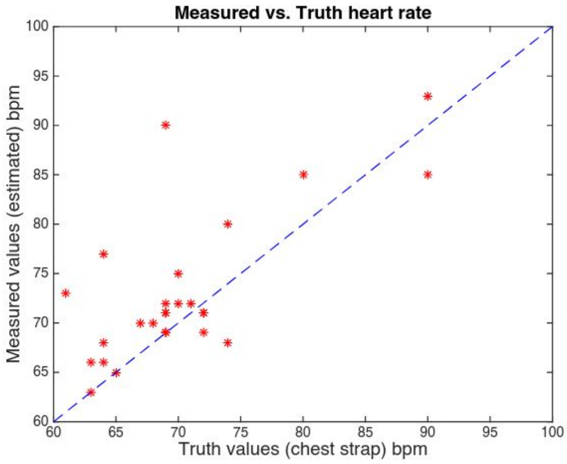


Figure 14: Measured vs truth heart rate

For better evaluation we also use the BlandAltman plot to represent the spread and bias between prediction and labeled values .

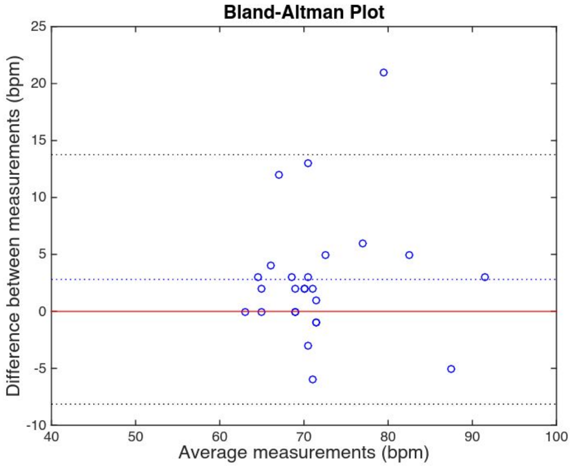


Figure 15: Bland-Altman Plot

The average absolute error between our prediction and is 4 bpm approx.
The error of fitness devices is around 3 bpm.
Therefore, the method produced by this paper is very accurate and on head to head level with most of the fitness bands available.

The best case for performance is achieved when the user and camera are static and therefore the user's vital sign is constant. Once the user's vital sign is constant all of the energy within the vital sign signal is at one frequency value; this produces a frequency response with a differentiating peak.

If the user's vital sign is extremely variable, then info within the frequency response can have degraded information. As a consequence, there will be no clean single peak that coincides with the user's vital sign. the answer to the present downside is accessible with a camera that contains a higher frame rate.

This is able to offer us higher temporal resolution and enough samples to investigate the short-time info within the signal, permitting us to capture changes in an exceedingly varied vital sign signal.

The drawback of our algorithm is that it is very computationally expensive, and without using cuda it crashes on mid-range processors(i5) .
Therefore if implementation in real-world needs to be done, we will need a special chipset architecture and good optimization.

Performance of the Model:

After Hyperparameter tuning, the optimized model achieved an accuracy of 75.83%, which is greater than the human baseline accuracy for such tasks. Our implementation has a further advantage of being able to identify early signs of drowsiness in real time giving the driver ample time to stop the car in time for rest.

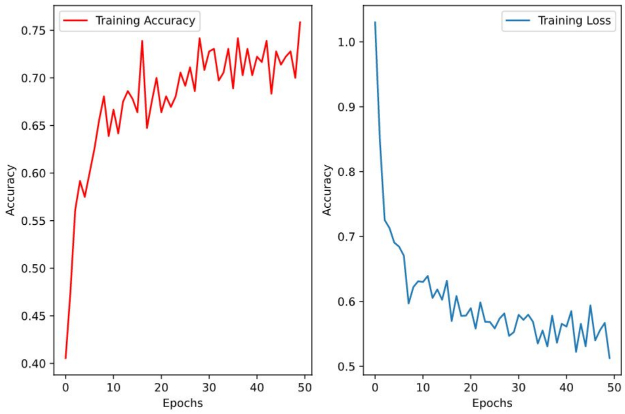
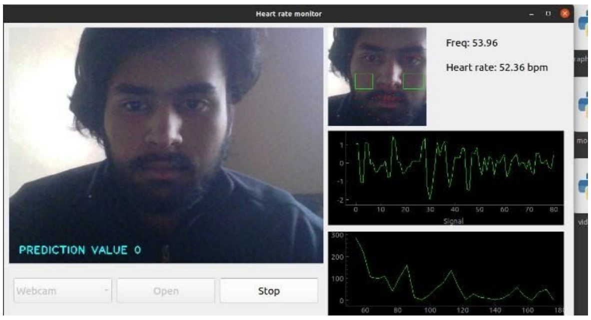


Figure 16: Model Training Metrics

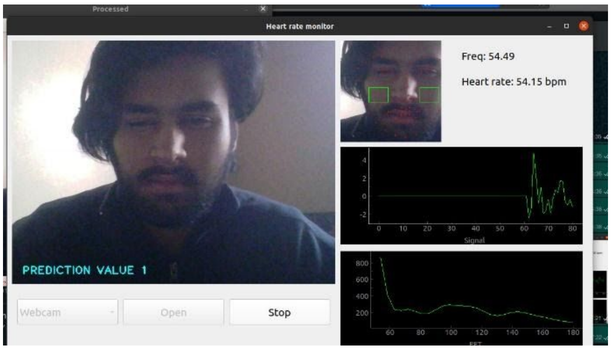

Heart rate monitor

Freq: 53.96

Heart rate: 52.36 bpm

PREDICTION VALUE 0

Figure 17 : Result with active driver


Heart rate monitor

Freq: 54.49

Heart rate: 54.15 bpm

PREDICTION VALUE 1

Figure 18: result with sleepy driver

VI. Limitation

The main problem with the model is the scarcity of generalized data. When training the model on the whole dataset, there was data available on 60 persons only. This leads the model to overfit on these features, even though steps were taken to address this problem by using Feature Normalization and Dropout Regularization, the problem still exists

VII. Future Scope

In the future we would like to increase the size of the dataset on which the model is trained on and include a wider range of individuals of different ages and ethnicities to improve the robustness of the model. Also the videos used in the dataset were shot in good lighting conditions which is not indicative of night-time driving where the model will be mainly used. As such we would use videos shot in low-light conditions for better simulation purposes.
Also instead of using a unidirectional LSTM model, we would use a HM-LSTM (Hidden Markov Models-LSTM) to better track the long-range dependencies as this has been proven to give better results [1] .

VIII. Conclusion

In conclusion, this paper showcased that on evaluation of the final model there is only a delay of one second and error of margin of 4bpm for predicting the heart rate . Also, after Hyperparameter tuning, the optimized model achieved an accuracy of 75.83%, which is greater than the human baseline accuracy for such tasks.

References

[1] Yu, J., Chen, Z., Zhu, Y., Chen, Y., Kong, L., & Li, M. (2016). Fine-grained abnormal driving behaviors detection and identification with smartphones. *IEEE transactions on mobile computing*, *16*(8), 2198-2212.

[2] Samiee, S., Azadi, S., Kazemi, R., Nahvi, A., & Eichberger, A. (2014). Data fusion to develop a driver drowsiness detection system with robustness to signal loss. *Sensors*, *14*(9), 17832-17847.

[3] Miyajima, C., & Takeda, K. (2016). Driver-behavior modeling using on-road driving data: A new application for behavior signal processing. *IEEE Signal Processing Magazine*, *33*(6), 14-21.

[4] Zhao, L., Wang, Z., Wang, X., & Liu, Q. (2017). Driver drowsiness detection using facial dynamic fusion information and a DBN. *IET Intelligent Transport Systems*, *12*(2), 127-133.

[5] Verma, B., & Choudhary, A. (2018, September). Deep Learning Based Real-Time Driver Emotion Monitoring. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 1-6). IEEE.

[6] Mandal, B., Li, L., Wang, G. S., & Lin, J. (2016). Towards detection of bus driver fatigue based on robust visual analysis of eye state. *IEEE Transactions on Intelligent Transportation Systems*, *18*(3), 545-557.

[7] Xing, Y., Lv, C., Zhang, Z., Wang, H., Na, X., Cao, D., ... & Wang, F. Y. (2017). Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition. *IEEE Transactions on Computational Social Systems*, *5*(1), 95-108.

[8] https://sites.google.com/view/utarldd/home

[9] http://dlib.net/

[10] https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

[11] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013, December). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops* (pp. 397-403). IEEE.

[12] Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, *47*, 3-18.

[13] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 896-903).

[14] Ghoddoosian, R., Galib, M., & Athitsos, V. (2019). A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0-0).

[15] Public Benchmark Dataset for Testing rPPG Algorithm Performance 22.07.2020, by Wouter Hoffman, Daniel Lakens

[16] OpenCV 2 for python 3 https://docs.opencv.org/

[17] M. P. Tarvainen, P. O. Ranta-aho, and P. A. Karjalainen, "An advanced detrending method with application to HRV analysis," Biomedical Engineering, IEEE Transactions on, vol. 49, pp. 172-175, 2002.

[18] D. Cochran, "A consequence of signal normalization in spectrum analysis," in Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, 1988, pp. 2388-2391 vol.4.

[19] H. Rahman, M. U. Ahmed, and S. Begum, "Non-contact Physiological Parameters Extraction using Camera," in The 1st Workshop on Embedded Sensor Systems for Health through Internet of Things (ESS-H IoT), Oct., 2015.

[20] Ghoddoosian, R., Galib, M., & Athitsos, V. (2019). A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0-0).

[21] Smoothness Prior Approach to Removing Nonlinear Trends from Signals by Yu Jiang Wang