Detecting Drowsiness Using Face Information

Manikishore Medam
Dept of Computer Science
Drexel University
Philadelphia, USA
mm5224@drexel.edu

Pooja Nanda
Dept of Computer Science
Drexel University
Philadelphia, USA
pn364@drexel.edu

Shyam Senthil Nathan
Dept of Computer Science
Drexel University
Philadelphia, USA
ss4982@drexel.edu

***Abstract*—Driver's drowsiness accounts to a prominent number of accidents worldwide and effectively identifying when drowsiness sets in in our faces will help in averting such risks. A lot of approaches have been followed for detecting drowsiness in the face. In this current approach, we have used facial expressions, specifically the width of the eyes and the mouth and their ratios, considering the fact that the height of the eye will get reduced when the person is drowsy, and yawning increases the width of the mouth, and used this information for creating a model that can identify whether a person is alert or drowsy. We have used feature selection to identify and capture the face and the required features and then built various models – logistic regression, CNN and LSTM, to determine drowsiness. This can be applied in various scenarios and has multiple applications, as in detecting drowsy drivers, alertness checking applications, and detecting drowsy workers in heavy machinery factories to avoid accidents.**

***Index Terms*—Drowsiness Detection, Feature Selection, Logistic Regression, Convolutional Neural Network, LSTM, Deep Learning.**

I. Introduction

Drowsiness is an intermediate state between alertness and sleep. This unassuming state of our consciousness can become very dangerous in certain circumstances. Road crashes and accidents are a prominent cause of injury and death among humans, and according to 2015 data from the World Health Organization, road traffic injuries resulted in approximately 1.25 million deaths worldwide, i.e. approximately every 25 seconds an individual will experience a fatal crash[1]. While there are many reasons for accidents to occur, driver drowsiness accounts for approximately 100,000 accidents per year in the United States alone as reported by The American National Highway Traffic Safety Administration (NHTSA)[2]. Seems ironic but doctors, mostly surgeons, after long shift hours are highly prone to road accidents due to drowsiness. Detecting drowsiness can help us avoid such untoward incidents. There are other areas where this can come in handy. Drowsiness in industrial scenarios may result in improper use of machinery, which may lead to reduced efficiencies, and in some cases, even injuries that may even be fatal. Classrooms are a classic use-case, where the teacher might want to gauge the interest or attention levels of the students and may use the information to improve their presentations.

Many researchers are working on techniques to estimate drowsiness. Several estimation techniques have been proposed, like subjective assessment, sensorimotor indicators, physiological features, and driving behavior and performance. This paper uses the physiological features like eyes and mouth and determines if a person is drowsy or not. Facial expressions are recorded from the user, prominent features that aid in drowsiness detection like eyes, mouth, orientation are extracted and the dimensions and ratios of these features are used to concluded if the person is drowsy or not.

II. Background

There has been a considerable amount of work done in this field as discussed in the Related Work section. We are basing our work on the following hypotheses:

- A drowsy person's eye will be shut, the height of the open part will be lesser than when they are alert.
- They will yawn occasionally, which will cause the mouth to be open
- Both of these occurrences combined can help predict the alertness state even more efficiently.

We have obtained data-sets of images with both alert and drowsy faces as explained in the Implementation section. We are basing our calculations of the state on the levels of openness of the eyes

and the mouth, and creating models to predict the results given new data. Some of the algorithms that we have used in creating the training data for the models is borrowed from the paper describing the UTA-RLDD dataset [3] that we have obtained our dataset from.

III. Related Work

Initial studies or approaches that were used to detect drowsiness used expensive hardware such as EEGs[8]. Several later approaches have been applied in detecting drowsiness in people, but most of them have been limited to situations involving driving[3], where there was a plethora of data available including position of hands on steering wheels, movement of steering wheels, amount of acceleration/braking, etc. In studies that were not restricted by the use-case of driving, there have been different approaches to detecting drowsiness. One of them is extracting eye features and determining based on various parameters such as openness and squinting to detect if the face is drowsy[4]. There have also been studies that have determined drowsiness taking into account the blinks made, number of blinks and their frequency[5][6]. Face expressions have also been used to detect drowsiness in faces[7]. While these earlier computer-vision based approaches to detecting drowsiness have been based on one of several facial features, our approach includes a combination of features and parameters related to the eyes, pupils and mouth, which we hope will give better results. We also use multiple classification methods to identify the strengths and weaknesses of each approach for different use cases and come up with the most efficient approach.

IV. Implementation

We have collected our data from UTA – Real Life Drowsiness Data-set provided by the University of Texas at Arlington in the form of a series of videos captured on 60 participants, each participant providing three classes of facial expressions – alert, low vigilance and drowsy. Each video was classified using this information and was used for our models.

A. Data Processing

The data-set comprises around 30 hours of videos recorded on around 60 unique participants. For each video, we used openCV to extract 1 frame every 2 seconds from start until the end of the video. Each video was approximately 8 minutes long, so we extracted around 240 frames per video, contributing a good amount of data that would be well-suited to handle data dominant algorithms like CNN and LSTM.

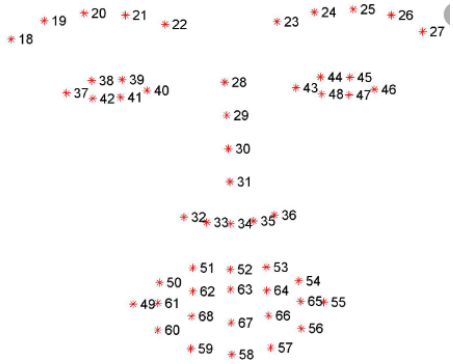

Fig. 1. Landmarks from OpenCV [10]

B. Feature Extraction

We then used openCV to extract facial features from the image frames. There were 68 total landmarks per frame from which we collected landmarks related to the eyes and mouth only (Points p37–p68). These were the important data points that reflect the key points in face recognition that we were interested in - the eyes and the mouth. As shown, landmark in Fig. 1, points p37-p48 were considered for the eyes, and p49 – p68 for the mouth. From the extracted points, we calculated the aspect ratio of the eyes (EAR), aspect ratio of the mouth (MAR), circularity of the pupil (CIR), and the ratio between EAR and MAR (EMR).

1) EAR: As shown in Fig. 1, points 37 through 42 and 43 through 48 represent the two eyes. We calculated EAR as

$$EAR = \frac{||p38 - p42|| + ||p39 - p41||}{2|\,|p37 - p40||}$$

for the left eye. The same calculations were also used for the right eye with the corresponding points, and for all input frames. Calculating EAR as above gives an estimate of the extent to which the eyes are closed. This value being close to zero correspond to the indication of closure of eye and a bigger value assures that eyes are open. The EAR is used in

other applications too like calculating the number of blinks and more.

2) MAR: The mouth is represented as 8-coordinates in the output from the openCV feature extraction method, these points located on crucial edges help in determining the width and height of the mouth when closed and opened. We used the formula

$$\frac{p52 - p58}{p55 - p49}$$

to calculate MAR. We hypothesize that people start yawning more frequently when they are drowsy, and the level to which the mouth is open is the best indicator to detect yawning. This means that a higher MAR means that the mouth is open and the MAR becomes 0 or close to 0 when the mouth is fully closed. Since we will be combining this along with readings of the eyes, artifacts such as talking and other mouth movement will not affect our calculations.

3) CIR: The circularity of the pupil is calculated by using the formula:

$$CIR = \frac{4 * \pi * Area}{Perimeter^2}$$

where

$$Area = \Big(\frac{Distance(p38, p41)}{2}\Big)^2 * \pi$$

and

$$\begin{aligned} Perimeter = {} & Distance(p37, p38) + \\ & Distance(p38, p39) + Distance(p39, p40) + \\ & Distance(p40, p41) + Distance(p41, p42) + \\ & Distance(p42, p37) \end{aligned} \quad (1)$$

for the left eye, and with the corresponding points for the right eye.

CIR helps us in estimating the area of the eye. CIR also helps in determining how much is the eye is closed, a smaller value indicates that the eye is closed indicating that the user is drowsy and a larger value indicates that the eye is open and the person is alert.

4) EMR: Finally, we have EMR, the ratio between EAR and MAR. We hypothesize that if EAR changes, then MAR will have an opposite directional change in case of drowsiness, as the eyes will droop, and the mouth will yawn, as we get a steady rise in EAR and fall in MAR, the value of EMR decreases indicating that the person is getting drowsy.

C. Modelling the features

The calculated values of EAR, MAR, CIR and MAR are then used to develop a model that can use this information, train on it and estimate the incoming image frame if drowsy or not. Before building the models, these ratios are normalized and then are fed to the model. Normalization helps in distributing the data to a known range and will improve the performance of predicting the label. We normalized the features only related to alert video frames to get a estimate of the range of each feature distribution, to essentially increase the distance between a drowsy input and alert data, thus improving the model accuracy.

1) Logistic Regression: Logistic regression is a statistical model that establishes a relation between single or multivariate variables to an output variable. It is prominent algorithm that uses the input variables and tries to estimate an equation or dependency of the output variable on such input variables. Although it take only a single input at a time unlike neural networks that takes weights, bias and feedback from other inputs. Since logistic regression cannot account for sequential data, we averaged data from 3 subsequent frames to improve accuracy of our prediction.

2) Convolutional Neural Networks: CNNs are a class of neural networks, that are regularized versions of multi-layer perceptrons. CNNs have proven very effective primarily in image processing and classification. They take an input, assigns weights and biases to the perceptrons and learn to estimate the output. We have used a CNN to model the features and predict the label for each frame - drowsy or alert. We are classifying the data based on the feature rations calculated previously. We have used 1D-CNN available in Keras library to implement and used 5 layer approach, sigmoid and relu as

activation parameters to build the network. The layers that we used in our CNN is as below:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_4 (Conv1D)            (None, 6, 64)             256
_________________________________________________________________
flatten_5 (Flatten)          (None, 384)               0
_________________________________________________________________
dense_15 (Dense)             (None, 32)                12320
_________________________________________________________________
dense_16 (Dense)             (None, 16)                528
_________________________________________________________________
dropout_6 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 17
=================================================================
Total params: 13,121
Trainable params: 13,121
Non-trainable params: 0
_________________________________________________________________
```

3) Long Short-Term Memory Network (LSTM): LSTMs are a special kind of recurring neural network that have a capability of learning long term dependencies. Each node in LSTM consists of an input gate, output gate and hidden state, and an additional gate called forget gate added to the structure. These gates help in learning from the data and keeping only relevant knowledge and discarding the others. The cell state act as a framework or medium that transfers relative knowledge all down the sequence chain. For our experiment, we converted the frames into groups. Each group was sent through a fully connected layer with 1024 hidden units using the sigmoid activation function. The next layer is our LSTM layer with 512 hidden units followed by 3 more layers until the final output layer.

D. Results

Once we have trained all our models with the data-set that we obtained as described above, we tested our models with input from the original UTA-RLDD data-set that we did not use for the training of the models. The results that we obtained are tabulated below.

For the models we have used in classifying the data, CNN and Logistic regression gave better accuracies compared to LSTM model. Accuracy of Logistic regression shows that each aspect ratio plays crucial role and detection of drowsiness is directly dependent on these ratios. CNN proved again that it's best suitable for image classification while interestingly LSTM had lower accuracy than the rest. This might be a direct impact of our choice of layers. Neural networks took long time

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 6, 1024)           9216
_________________________________________________________________
lstm_2 (LSTM)                (None, 6, 512)            3147776
_________________________________________________________________
flatten_6 (Flatten)          (None, 3072)              0
_________________________________________________________________
dense_19 (Dense)             (None, 216)               663768
_________________________________________________________________
dense_20 (Dense)             (None, 32)                6944
_________________________________________________________________
dropout_7 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_21 (Dense)             (None, 16)                528
_________________________________________________________________
dropout_8 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_22 (Dense)             (None, 1)                 17
=================================================================
Total params: 3,828,249
Trainable params: 3,828,249
Non-trainable params: 0
_________________________________________________________________
```

to run to train and test the model, changing the depth of the network significantly increased the run time. Logistic regression also produced surprisingly good predictions, comparable to the results from the CNN.

TABLE I
ACCURACIES FROM MODELS IMPLEMENTED

Model Name	***Accuracy***
Convolutional Neural Network	0.7708
LSTM	0.5234
Logistic Regression	0.7786

Apart from testing with the data from the UTA-RLDD, we also implemented a system where we could feed live video data from our webcams and classify our current state. The results were surprisingly good for the CNN and logistic regression models, but the LSTM network struggled to identify the correct states. A sampling of our webcam results can be shown in the below figure.

V. FUTURE IMPROVEMENTS

We have collected frames only from 8 participants, and 1 frame every 2 seconds. We can drastically improve the data-set by collecting the frames from all the 60 participants, and multiple frames

Fig. 2. Labelled Output Images

each second, to build a better model, given enough processing power. We are also only considering only fully alert and fully drowsy states for training, considering states in-between might improve the results, and might also help us to determine various levels of alertness. Finally, there are a number of classifiers that can also be tried, such as Naive Bayes classifiers, random forest algorithm, and decision trees, that might produce different results. Even the algorithms that we have used can produce different result by varying a few parameters and layers.

VI. Conclusion

We have thus created a system that can identify if a given face is in a drowsy or an alert state. While we have achieved considerable levels of accuracy, this is still not enough to be applied in real-time applications. We do believe though that this approach can be tweaked to provide better results in the future, and can be applied in real-world applications such as in driving to prevent accidents, and in classroom and office settings to guage interest levels.

References

[1] https://www.cdc.gov/features/dsdrowsydriving/index.html

[2] https://www.nsc.org/road-safety/safety-topics/fatigued-driving

[3] Reza Ghoddoosian, Marnim Galib, Vassilis Athitsos. 2019. A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection.

[4] Naurois, Charlotte Jacobé de, et al. “Detection and Prediction of Driver Drowsiness Using Artificial Neural Network Models.” Accident Analysis Prevention, vol. 126, 2017, pp. 95–104.

[5] J. Ahmed, J. Li, S. A. Khan and R. A. Shaikh, ”Eye behaviour based drowsiness Detection System,” 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing

[6] W. Zhang, B. Cheng and Y. Lin, ”Driver drowsiness recognition based on computer vision technology,” in Tsinghua Science and Technology, vol. 17, no. 3, pp. 354-362, June 2012, doi: 10.1109/TST.2012.6216768

[7] M. Chakraborty and A. N. H. Aoyon, ”Implementation of Computer Vision to detect driver fatigue or drowsiness to reduce the chances of vehicle accident,” 2014 International Conference on Electrical Engineering and Information Communication Technology, Dhaka, 2014, pp. 1-5, doi: 10.1109/ICEEICT.2014.6919054.

[8] M. A. Assari and M. Rahmati, ”Driver drowsiness detection using face expression recognition,” 2011 IEEE Inter-national Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, 2011, pp. 337-341, doi: 10.1109/ICSIPA.2011.6144162.

[9] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/