Driver Drowsiness Detection using Artificial Intelligence

Kareem Atif Mohamed Bakli[1]
Electrical Engineering
Ottawa University
kbakl031@uottawa.ca

Mohamed Khaled Ramadan[2]
Electrical Engineering
Ottawa University
meles043@uottawa.ca

Mostafa Mahmoud Nofal[3]
Electrical Engineering
Ottawa University
mnofa091@uottawa.ca

Gehad Hisham Hassan[4]
Electrical Engineering
Ottawa University
gsoma101@uottawa.ca

Kareem Khaled Moner Waly[5]
Electrical Engineering
Ottawa University
kwaly008@uottawa.ca

***Abstract*—Driver Drowsiness is considered as a major reason for car accidents all over the world on the road. In this study, we introduce three different approaches to detect driver drowsiness. The first approach of the three approaches is statistical approach using dlib face recognition library, the second approach uses several machine learning models which is KNN, Random Forest, and SVC on a tabular dataset, and lastly, we use deep learning approach, which is CNN on images dataset. These approaches are applied on these 2 different datasets, this helps us to detect whether the driver is awake or sleepy.**

Keywords— Driver Drowsiness, Face Recognition, Computer Vision, Convolution Neural Network

I. Introduction

According to the National Safety Console (NSC), more than 100,000 cars crashes occur in result of drowsiness, from those crashes, about 71,000 people got injured, and 1,550 died. The scariest part about drowsiness is that can be as small as a little state of unconsciousness, this small amount of time can lead the driver to lose attention and control of the driving, which will not only cause the driver being injured but also other people. In our project, we try to decrease the number of accidents that happened because of drowsiness, by predict whether the face of the driver shows signs of being drowsy or awake and alarming him if he shows those signs before it is too late.

There are obvious signs that shows if the driver is drowsy or not, this includes:

- The driver is yawning,
- The eyes of the driver are barely opened.

For this work, using the pervious information, we used three paths that can lead us to reach our goal that is detecting the state of the drivers.

In our work, we develop three different approaches to detect drowsiness detection for the driver:

A. Statistical Model

In this approach, we depend on the facial features, these features are measured by two ways, one of them is whether the eyes are closed or opened by detecting the points of the eyes. And also know if the driver is yawning or not by calculate the distant between the lips. In this approach we and normal reading of the user facial expression as the feature, and the true label represents the state of the user as drowsy or not. And we chose the random forest as our champion model.

predict the state of the user by passing a certain threshold that we implemented manually.

B. Statistical Model with Machine Learning

This approach used three different machine learning algorithms, which are KNN, Random Forest, and SVC, to decide if the user is drowsy or not, by feeding the three models that were trained by certain dataset [1] that includes both normalized and normal reading of the user facial expression as the feature, and the true label represent the state of the user as drowsy or not. And we chose the random forest as our champion model.

C. Deep Learning Model using CNN

The DL Model, we used Convolution Neuron network (CNN), and the model was trained on drowses dataset [2], then the model was applied using computer vision approach and took the whole face and predicted if the user has one of these four states: opened eyes, closed eyes, yawned and not yawned. From the prediction we tried to estimate the state of the user of drowsiness.

II. System Architecture

A. DataSet

We used two datasets, one of the data sets is CSV file used for Machine learning Classifier (Random Forest) is generated from real time the UTA dataset consists of 180 RGB videos of 60 unique participants. Each video is around 10 minutes long, and is labeled as belonging to one of three classes: alert (labeled as 0), low vigilant (labeled as 5) and drowsy (labeled as 10).

Figure 1: test particular in the dataset

The second dataset is for yawned, non-yawned, opened and closed eyes dataset from Kaggle website. [second dataset]

B. Feature extraction

In the first two approaches, we used camera to take the photo of the driver face and used the dlib library to turn the image to turn the image into points figure 2, that we could use in our calculation like figure1.

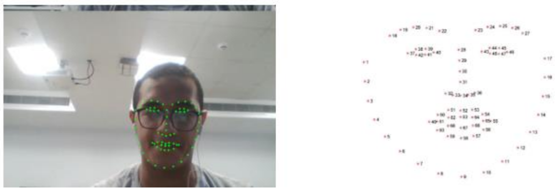

Figure 2: test case

C. Mathematical calculations

We computed four mathematical formulas, which are:
Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), Pupil Circularity, mouth over eye ratio.

The Eye Aspect Ratio (EAR) is the ratio of distances between the vertical eye landmarks and the distances between the horizontal eye landmarks. The return value of the eye aspect ratio will be approximately constant when the eye is open. Then the value will then rapidly decrease towards zero during a blink. If the eye is closed, the ratio will again remain approximately constant, but will be much smaller than the ratio when the eye is opened.

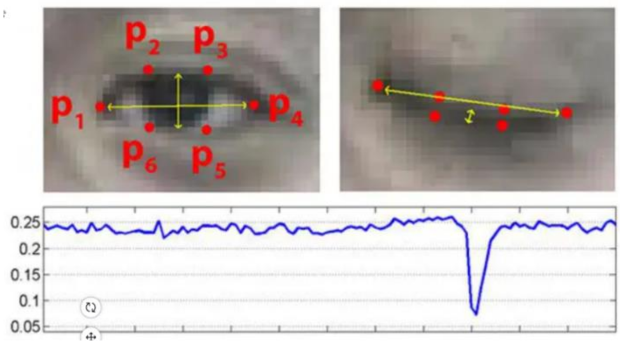

Figure 3: Eye Aspect Ration

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

Figure 4: EAR of one eye

The Mouth Aspect Ratio (MAR) is the output of calculating the distance between the mouth lips and creating a certain threshold that indicate if the person is yawning or not.

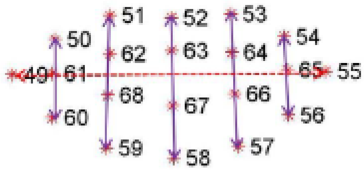

Figure 5: Mouth Aspect Ratio

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

Figure 6: Mouth Aspect Ration formula

The Pupil Circularity (PUC) extends the EAR formula, but it depends on the changing of the pupil of the eyes, it is A further indicator and probably more accurate indicator

$$PUC = \frac{4 * \pi * Area}{perimeter^2}, \qquad Area = (\frac{Distance(p2, p5)}{2})^2 * \pi$$

Figure 7: PUC formula

There is another indicator which is Mouth Over Eye Ratio (MOE), with is MAR divided by EAR.

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

Figure 8: MOE formula

D. Calibrating

In this part we scaled the ratio of each calculation because we want our approach to be more generalized and not biased towards specific kinds of faces, as the faces features of people changes, by calculated the mean and variance of the face before starting with the detection.

III. MODEL

A. Statistical Model

In the Statistical model we first using the dlib library we turn the image from the camera and to point that will be first Calibrated so we could calculate the mean and variance of the face to normalize the measurement of the face and then we will start calculate the state of the using the normalize image and predicting the state of the driver using the normalized EAR measurement and depending whether the threshold was passed or not the state of the user would determine.

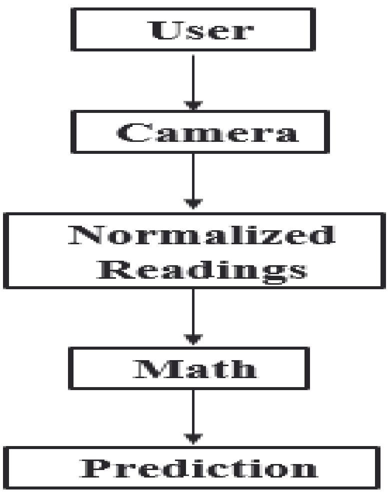

Figure 9: Statistical Model pipeline

The figure below shows the predation from the model

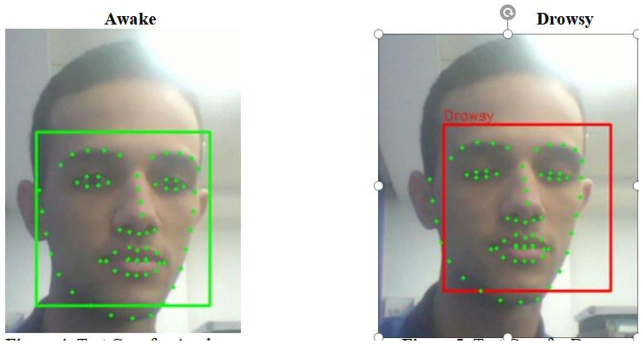

Figure 10: Test Case

B. Statistical Model with Machine Learning

In this model we used both the normalized and normal reading that was generated like the previous model but we used the help of machine learning instead of our own calculation to detriment the state of the user but we had we first used the csv dataset that had both normalized and normal reading as it’s feature and three states : normal ,drowse, and alert and we chose 3 different model: KNN, Random Forest, and SVC from the these three model the Random Forest was our champion model was the Random Forest champion model from these three and that happened by training and testing on the dataset, then we applied the now trained random forest model to and feed it with readings and get the prediction

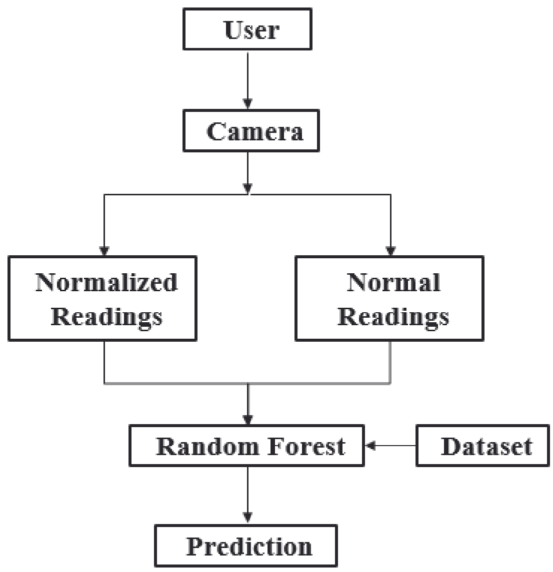

Figure 11: Model pipeline

The Machine Learning pipeline

The figure below shows the predation from the model

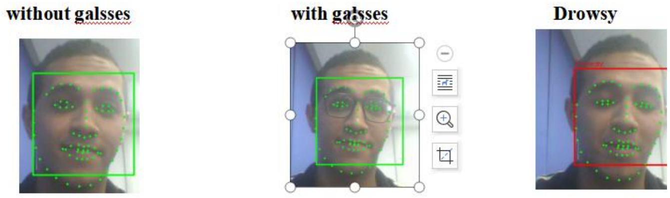

Figure 12: Test Case

C. Deep Learning Model using CNN

In this model we got away from the dlib library and tried another approach which was using the power of deep learning we trained a CNN on the open closed yawn Kaggle dataset the architect of the model is like the figure blow

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 143, 143, 256)     7168
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 71, 71, 256)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 69, 69, 128)       295040
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 34, 34, 128)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        73792
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 14, 14, 32)        18464
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 1568)              0
_________________________________________________________________
dropout (Dropout)            (None, 1568)              0
_________________________________________________________________
dense (Dense)                (None, 64)                100416
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 260
=================================================================
Total params: 495,140
Trainable params: 495,140
Non-trainable params: 0
```

Figure 13: Deep learning Model

And we saved the trained model in a .h5 model and then we used OpenCV and haar cascade to detect the image and then we applied the same prepossessing that we used in model and sent it to the model and finally we applied Argmax to predict which from the four states the driver is having.

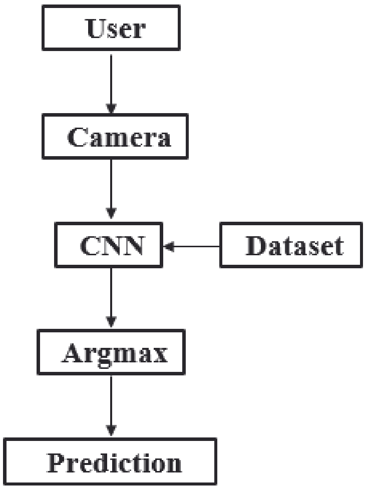

Figure 14: Deep learning Model

The Deep Learning pipelines

The figure below shows the predation from the model

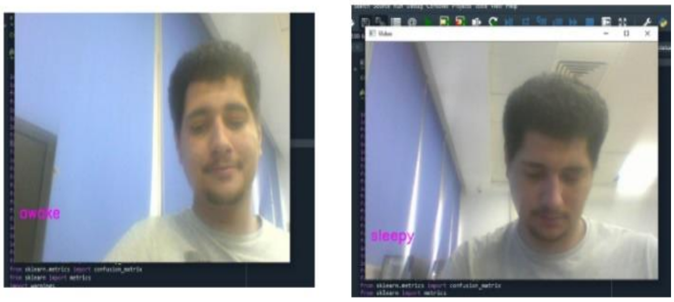

Figure 15: Test Case

IV. PERMOANCE EVALUATION

	RANDOM FOREST CLASSIFIER	SVC	K-NEIGHBORS CLASSIFIER
PRECISION	**0.83**	**0.7**	**0.76**
RECALL	**0.87**	**0.84**	**0.83**
ACCURACY	**0.81**	**0.7**	**0.75**
F1-SCORE	**0.85**	**0.76**	**0.79**

Figure 16: Model Evalution

We implement three Models RandomForestClassifier(), SVC() and K-NeighborsClassifier() and found that the best Model according to the accuracy was RandomForestClassifier(), then the K-NeighborsClassifier() and the worst one was SVC.

V. Summary and Conclusion

Statistical model accuracy is not good because it depends on the eye aspect ratio. This ratio is not constant and differs from one to another. Our second approach is a statistical model with machine learning. It is better than the first approach because it doesn't depend on just eye aspect ratio but also on mouth aspect ratio, pupil circularity (PUC) and mouth over eye ratio. In the final approach, we implemented a deep learning model. Its accuracy was the best compared with previous models with an accuracy of 97.1% but in live inference statistical with machine learning was the best.

VI. Future Work

To improve the performance of the deep learning model, we need to apply more data. In addition to upgrading the camera to capture better live snaps. We will consider implementing other models like Long Short Term Memory (LSTM) neural network. Also, we need to add more test cases at night with low light.

VII. References

[1] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som, and H. A. Ameen, "Drowsiness Detection System using Eye Aspect Ratio Technique," 2020 IEEE Student Conference on Research and Development (SCOReD). IEEE, Sep. 27, 2020.

[2] T. V. N. S. R. Sri Mounika, P. H. Phanindra, N. V. V. N. Sai Charan, Y. Kranthi Kumar Reddy, and S. Govindu, "Driver Drowsiness Detection Using Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), and Driver Distraction Using Head Pose Estimation," ICT Systems and Sustainability. Springer Nature Singapore, pp. 619–627, 2022.

[3] S. Dey, S. A. Chowdhury, S. Sultana, Md. A. Hossain, M. Dey, and S. K. Das, "Real Time Driver Fatigue Detection Based on Facial Behaviour along with Machine Learning Approaches," 2019 IEEE International Conference on Signal Processing, Information,Communication & Systems (SPICSCON). IEEE, Nov. 2019.

DataSet References:

[4] https://github.com/hackenjoe/Advanced_Drowsiness_Detection/blob/main/data_drowsiness/prepar ed/totalwithmaininfo.csv

[5] https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset

[6] https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new

Figure References:

[7] Figure [1]: Test particular in the dataset https://assets-global.website files.com/5d7b77b063a9066d83e1209c/61e9cf142e6e558b3990bdf2_UTA-RLDD.png

[8] Figure [2]: test Case

[9] Figure [3]: Eye landmark open and close https://ars.els-cdn.com/content/image/1-s2.0-S0957417420303298-gr3.jpg

[10] Figure [4,5,6,7,8]: Equations https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706

[11] Figure [9,11,14] : Pipeline Handmade

[12] Figure [10,12,15] : Test Case

[13] Figure [16]: Model Evalution

[14] Figure [13]: Deep learning model