Student Care – Sleepiness Detection System

Shivayavashilaxmipriya S
Department of applied mathematics and computational science
Thiagarajar college of engineering
Madurai, Tamil Nadu
shivayavashi.s@gmail.com

Archana S
Department of applied mathematics and computational science
Thiagarajar college of engineering
Madurai, Tamil Nadu
archanas210603@gmail.com

Anitha. D, Assistant professor in Data Science
Department of applied mathematics and computational science
Thiagarajar college of engineering
Madurai, Tamil Nadu
anithad@tce.edu

Abstract— **Students need to deal with a lot of work like assignments and projects daily in their career path. They need to handle important days when the test or the project deadline is nearby. There may be situations when they have to work diligently and have to attend important class sessions. Students frequently experience daily drowsiness, sleep deprivation, and irregular sleep cycles during their schedules and work. They are particularly affected by the negative effects of sleep deprivation and daytime drowsiness, which can result in poorer grade point averages, an increased chance of academic failure, degraded learning, and decreased mood. The sleepy state of students can be determined by their eyes by classifying them into active, drowsy, and sleepy states. The student gets an alert if they are in a sleepy state and this will help them to manage their work by taking short breaks. The student sleepiness detection system is developed with the reference from works done for driver drowsiness detection systems as both focus on sleepiness detection. This paper focuses on developing a system-Student Care for detecting the sleepiness of students and alerting them. This will help them concentrate and take necessary breaks from the work. This system uses 3 Deep learning models developed with Dlib and OpenCV, 3-layer Convolutional Neural Networks (CNN), and 4-layer CNN that detects the state of the student and alerts them if they are in a sleepy state. The Dlib model uses Euclidean distance and Eye Aspect Ratio (EAR) for the detection of a sleepy state. The CNN models use eye images to classify the state of the student. Among all the models created, the 4-layer CNN model is determined to be the most accurate, with an accuracy rate of 96%.**

Keywords—Deep Learning, CNN, EAR, epochs

I. Introduction

A. Background

Students handle a lot of work in their day-to-day life. The drowsy state, tired state and attention state are all parts of the learning state, and drowsy state can accurately reflect how well students do in class. Deep learning plays a significant role in our lives. This deep learning model is developed to detect the sleepiness of students. This project is addressed for students whose daily workload includes numerous projects and assignments. They typically become worn out as a result of the constant academic work and assignments. Students tend to be sleepy in the classroom, primarily as a result of late-night homework assignments. The solution to this problem is to use deep learning to create a sleepiness detection model. The student's eye state is the focus of this model. Drowsiness is detected by observing the student's eye state and movement. The driver drowsiness system already addresses problems of this kind. The primary goal of this work is to keep the classroom alive by using deep learning to analyse eye movement and state to identify students who are asleep or are about to fall asleep. Since there are a limited number of sleepiness detection systems developed earlier, driver drowsiness detection is taken as reference for this project. To classify the eye status deep learning methods can be used namely AlexNet and GoogleNet. Both makeup and no makeup, close eye and open eye images are taken for experimentation. Along with that a support vector machine can also be used to classify the eye status for judging the drowsiness [1]. The position of the irises and the states of the eyes are tracked over time to estimate the amount of time it takes for the eyes to close and blink[2]. HOG-Histogram of Oriented Gradient is commonly used for face detection and Random Forest is used for classification [3]. The student's sleepiness can be detected through eye blinking based technique in which the eye status is monitored through the camera [4]. Detection of eye can also be done using cross correlation based online dynamic template matching technique combined with SVM and HOG [5]. Deep learning automatically performs feature extraction and modelling following data training, whereas machine learning needs data scientists or users to do it.

B. Existing methods

Table 1 – Existing methods

Ref	Methods used	Objective
[1]	AlexNet and GoogleNet	To detect drowsiness of drivers with the eyes using images consisting of makeup and no makeup images.
[2]	Template Matching, Eye Blinking, yawning based technique based and Artificial Neural Networks technique.	To propose an Advanced Driver Assistance System (ADAS) to detect drowsiness of drivers.
[3]	HOG in face detection and Random Forest for classification	To detect drowsiness of drivers to minimize road accidents

[4]	SVM with EEG and ECG monitoring.	To detect drowsiness with heart rate variability and frequencies.
[5]	Viola-Jones Algorithm for face and eye detection and SVM for classification. HOG and Local Binary Pattern (LBP) is used for preprocessing.	To classify the eye state as open and close to detect drowsiness.

Traditional machine learning models cannot solve some issues, but deep learning models can. Hence the Deep Learning models are considered for the development of sleepiness detection models. CNN models are most widely used to classify images more accurately. Running CNN models with more epochs and batches can help them classify the state more accurately. The preprocessing techniques include grayscaling, resizing images and binarizing labels. The paper aims to detect the sleepiness of students and to alert them if they are in a sleepy state. We developed 3 models that include Dlib and OpenCV library, 3-layer, and 4-layer CNN models that produced more accurate classification of the student's state. Our goal is to deliver the most accurate model for classification. The model with the high accuracy and lower loss is concluded to be the best among all the models developed.

II. LITERATURE SURVEY

A face conveys a lot of information about the state of a person. The frequency of blinking and yawning differs in the state of fatigue. A DriCare system has been developed which detects the driver's fatigue status with the yawning and blinking frequencies. The Kernelized Correlation (KCF) algorithm also used Histogram of oriented gradient features for face tracking (Deng & Wu, 2019).Drowsiness has been one of the major causes for driving accidents which can be detected using image processing techniques that used Viola-Jones Algorithm to detect the drowsiness in a driving simulator (Poursadeghiyan et al., 2018).The work has presented the methods of using Percentage of Eye Closure (PERCLOS) and EAR to measure the eye closure percentage and to measure the ratio between the height and width. The findings of prominent algorithms from the work were Support Vector Machine (SVM), Hidden Markov model (HMM), and CNN. The work study has proved that the highest accuracy of 99.74% was obtained with CNN (Ngxande et al., 2017). A study conducted with 12 participants was used to explore 18 features of eye movements that classified the state of drivers into sleepy and alert states. The Sequential Floating Forward Selection (SFFS) algorithm was used for feature selection. The face, eye and mouth were detected with the techniques including Hough Transform, Landmark Model Matching, Haar Algorithm, Adaboost, Gabor Filter and disparity map (Jin et al., 2013).The driver fatigue detection has been done with the non-intrusive methods Electro-oculogram(EOG) and Electro-encephalogram(EEG).The study used electrodes placed near the eyes of the subject to detect the EOG signals(Clement et al., 2015).A paper work has presented advanced computer vision and mobile technology using smartphones to monitor visual indicators of driver fatigue. The study was conducted with 20 drivers and the images were taken as input data. The face was detected using a Haar-like feature detector. The front camera of the phone captured the images of drivers and then the images were fed to the CPU which used Standard Sleepiness Scale (SSS) and Karolinska Sleepiness Scale (KSS) to measure the drowsiness (He et al., 2013). A study collected the images via webcam using C++ OpenCV. The image pixels were represented in RGB color space. The four different eye positions - straight, left, right and upward were used for K-means classification with the k value as 3. From the literature review it is observable that Convolutional Neural Networks performed well for sleepiness detection with higher accuracy. Convolutional neural networks (CNN or ConvNet) are a subclass of neural networks that are mostly utilized in image recognition applications. With no loss of information, its integrated convolutional layer reduces the high dimensionality of pictures. CNNs are therefore very well suited for sleepiness detection. This work aims to develop a highly accurate sleepiness detection model with CNN.

III. DATA COLLECTION AND REQUIREMENTS

Data collection is the first and most crucial step in the process of data analysis. Although it is very cool to be able to identify faces in photos or videos, more information is needed about the person's face, such as their position, whether their eyes are open or closed, whether they are looking up, and other details, in order to create powerful applications. For the first model, Dlib is used, which is a landmark's facial detector with pre-trained models. It is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face. For face detection, we use OpenCV(Open-Source Computer Vision Library) which is a free and open-source software library for machine learning and computer vision that uses a Haar cascade classifier to find things in images, no matter how big or small they are or where they are. This algorithm can run in real time and is not very complicated. The shape_predictor_68_face_landmarks.dat file is used to detect and predict the 68 landmarks on face. For the next 2 models developed, namely 3- & 4-layer CNN, we used a dataset called Yawn_eye_dataset_new dataset from Kaggle. This dataset consisted of 2900 images that were divided into 2 directories namely training and testing with 2467 images in training data and 433 images in testing data. The directories contained eyes opened, closed and mouth yawn and no yawn images. The eye images were used for the development of both of the CNN models.

Technologies used:

1.Kaggle kernel- It is used to build the CNN models for the sleepiness detection system.

2.Visual studio - It is used to build the 3 python models for the System.

3.Python 3.9.6

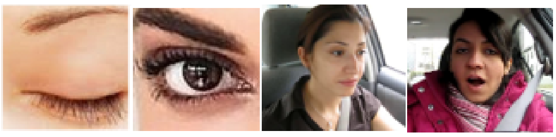


Fig 1: The sample images of dataset used for CNN model

IV. METHODOLOGY

The methodology used for the model development is shown in the below flowchart.

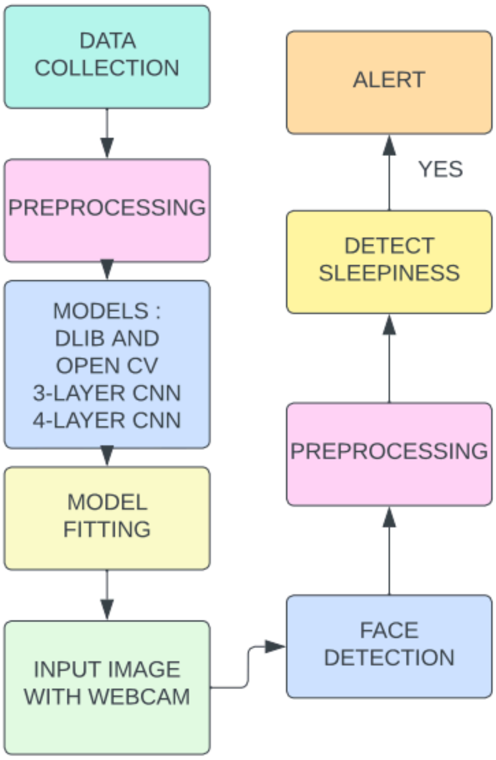


Fig 2: Methodology Framework

A. *Data Preprocessing*

Preprocessing techniques are used in face detection systems to increase detection time and minimize false positives. There should be a preprocessing phase that rejects a reasonable number of non-face windows.

I. *Grayscaling:*

The grayscaling is used to convert an image from a color space to shades of Gray. It varies between white and black color. It is used mainly for:
1.Dimensionality reduction:
Grayscale images are single dimensional whereas other color spaces are 3 dimensional.
2.Reducing model complexity
3.For many algorithms to work.
The OpenCV's cv2.cvtColor() function is used to convert RGB images into grayscale images.

II. *NumPy conversion:*

The Dlib face landmark detector will return a shape object containing the 68 (x, y)-coordinates of the facial landmark regions. Using the shape_to_np function, we can convert this object to a NumPy array.

III. *Binarize labels:*

In CNN models we use this function for preprocessing. The preprocessing is mainly done with the dataset collected, here to convert the multiclass labels into binary class labels we use labelbinarizer.It will convert a list into a matrix with precisely as many columns as there are distinct values in the input set.

IV. *Resizing the images*

For resizing the images, we use ImageDataGenerator which is used for getting the original data and transforming it on a random basis. It is also used to carry out the data augmentation where we can get the increment in the generalization of our model. It supports operations such as rotations, translations, shearing, scale changes and flips. In the 3-layer CNN model, we use it to rescale the images and also for grayscaling. In the 4-layer CNN model, we use it to rescale and horizontally flip the images and rotate it to 30 degrees.

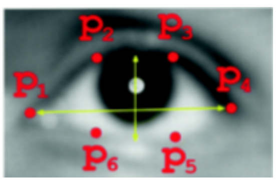
V. *Splitting the dataset*

The dataset is split into train and testing data before model building. The first model of CNN splits data into two batches of train and valid batch that uses the images in the 2 directories of our dataset. The 4-layer CNN model splits the entire dataset into a 70:30 ratio for training and testing the data.

B. *Model Development*

A. Dlib and OpenCV:

Here we use OpenCV and Dlib to build the model. The 68 landmarks in the face shown in figure 7 are detected first. Then the Euclidean distance among the landmarks of the eyes is computed and then the EAR ratio i.e., Eye Aspect Ratio shown in figure 3 is used to detect whether the eyes are open or close. The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks. If the ratio is greater than 0.25 then the student is said to be in an active state. If the ratio is between 0.21 and 0.25 then the user is in a drowsy state and else the person is in a sleepy state. The figures 4 , 5 and 6 shows the outputs of the detected faces with the classification.



$$EAR=[(P_2-P_6)+(P_3-P_5)] / [2(P_1-P_4)]$$

Fig 3: Eye Aspect Ratio

B. 3-layer CNN model:

First the model is built using the Kaggle kernel and downloaded as a h5 file and then the model is loaded in the python file created in the visual studio to detect the status of the person. The preprocessing is done with the ImageDataGenerator by resizing and grayscaling the images and then the model is built with the training data of 77 in each batch. The model developed is a sequential model with the operations of MaxPooling, Relu and SoftMax activations. The 3-layer model is developed with 30 epochs and then fitted and downloaded. For detecting the status of the person, first the face and the eyes must be detected, which can be done using the Haar cascade classifier which is generally used for the frontal, left and right eye. Then the face is read from the webcam and then converted to grayscale images for detecting the face and eyes. Based on the CNN model developed, the status of the eye is labeled as open and close based on the maximum probability of the class it belongs to. In every iteration of the while loop if the eyes are closed continuously the score is incremented and if the score value exceeds 15 then the person is alerted with the alarm sound. Else the score is zero and the alarm is not used. The figure 8 and 9 exhibits the output for the faces detected.

C. 4-layer CNN model:

First the model is built using Kaggle kernel and downloaded as a h5 file and then the model is loaded in the python file created in the visual studio to detect the status of the person. In the Kaggle kernel the 4-layer model is developed with 50 epochs with 43 as batch size and is then fitted and downloaded. The model is developed by splitting the data in 70:30 ratio for training and testing with MaxPooling operation Relu and SoftMax activations. Then we use a webcam to capture the face and detect the face and eyes using the Haar cascade classifier. The status of the right eye is detected and then the alarm is played if the count is greater than 10 here and the status is marked as closed else the eyes are open and the status is active. The figure 9 and 10 shows the output screens of the detected faces.

OUTPUT IMAGES

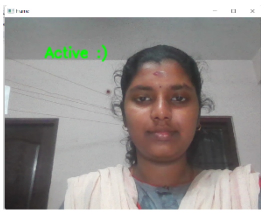


Fig 4: Active state

Fig 5: Drowsy State

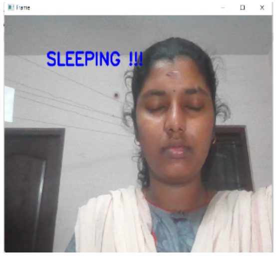


Fig :6 Sleeping State

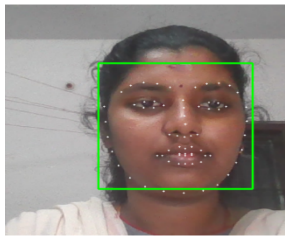


Fig 7: 68 points detection

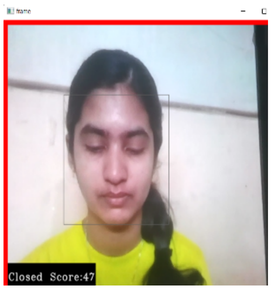


Fig 8: Sleeping State

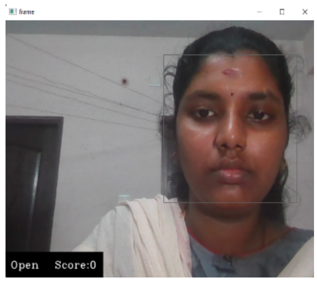


Fig 9: Active State

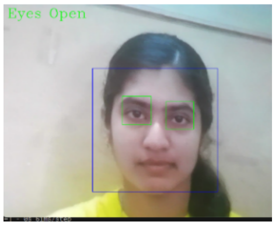


Fig 10: Sleeping State

Fig 11: Active State

V. RESULT AND DISCUSSION

A. Model Evaluation

In this section, the results obtained from all the models developed are denoted. The models developed were Dlib and OpenCV, 3-layer CNN and 4-layer CNN. The best model is evaluated with the accuracy as the performance metric.

B. Performance Analysis

I. ***Dlib model:***

Dib's shape detector is used to map the coordinates of the facial landmarks of the input video and drowsiness detected by monitoring aspect ratios of eyes and mouth. The maximum accuracy obtained for the Dlib model is 95%.

II. ***3-layer CNN model:***

The maximum accuracy obtained from the 3-layer Sequential CNN model is 99% for the training dataset and 95% for the testing data.

III. ***4-layer CNN model:***

The maximum accuracy obtained by the 4-layer Sequential CNN model for both the training and testing data is 97% and on an average the accuracy was 96%.

The figure 12 shows the training and testing accuracy of the model developed using 3-layer CNN. The 2nd part of the image shows the loss of the training and testing data used. We can infer that the training data shows more accuracy than the testing data. The figure 13 shows the training and testing accuracy of the 4-layer CNN.

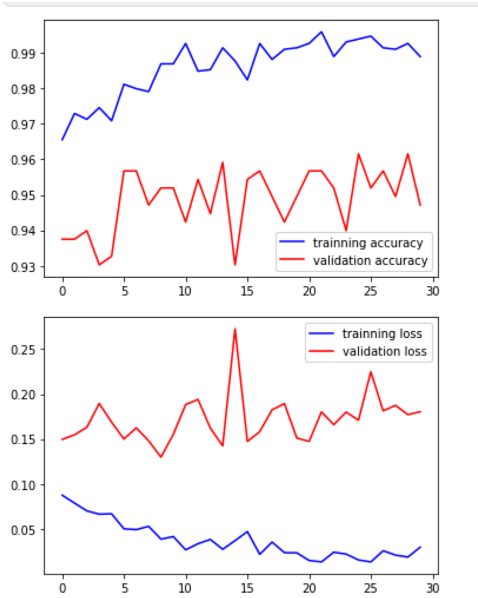


Fig 12: Accuracy graph for 3-layer CNN model

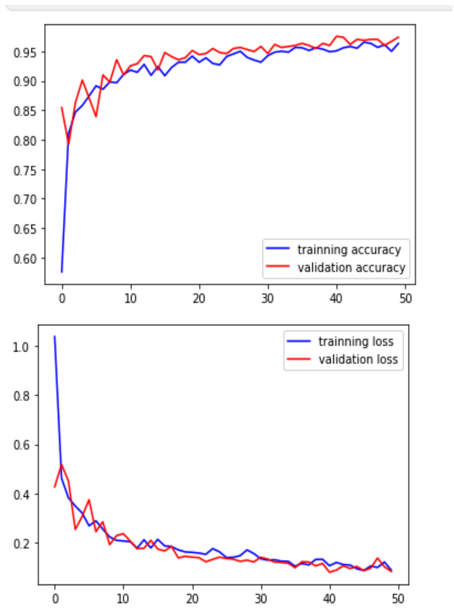


Fig 13: Accuracy graph for 4-layer CNN model

```
from sklearn.metrics import classification_report
print(classification_report(np.argmax(y_test, axis=1), prediction, target_names=labels_new))

              precision    recall  f1-score   support

        yawn       0.93      0.84      0.88        63
     no_yawn       0.88      0.95      0.91        74
      Closed       1.00      0.94      0.97       215
        Open       0.95      1.00      0.97       226

    accuracy                           0.96       578
   macro avg       0.94      0.93      0.93       578
weighted avg       0.96      0.96      0.95       578
```

Fig 14: The classification report for the CNN model

The classification report obtained for the 4-layer CNN is shown in the above figure with the recall of 0.94 for closed eyes.

The literature survey showed the highest accuracy obtained with CNN models. From the results obtained it is evident that the CNN models produced accurate results with high accuracy and lower loss.

Table 2 – Accuracy obtained from the models

Model	Accuracy
Dlib and OpenCV model	95%
3-layer CNN model	Training data-99% Testing data - 95%
4-layer CNN model	Training data-96% Testing data - 96%

VI. CONCLUSION

In this paper, we described the design and implementation of a system that uses eye movement and eye state to detect sleepiness in students and then sound an alarm to alert them if they are in a sleepy state. The primary component of this system is a camera that is utilized for the recording of facial landmarks. This paper tries to look at various algorithms and figure out the best ways to try to detect the student's sleepiness in the classroom. Algorithms like DLIB & OPENCV, 3-layer CNN and 4-layer CNN are used for this project. Based on the results of the experiments, our project is very effective at predicting the student's level of drowsiness by monitoring their eye movement and state. The model works well for images captured at good lighting conditions. In our analysis the 4-layer CNN model exhibits 96% accuracy. Thus, the 4-layer CNN model is the best model in our study. Moreover, we will implement our model to predict the sleepiness of students who wear spectacles in future. Our future works include setting a high quality camera that predicts the faces in the classroom atmosphere and evaluating it. .

Acknowledgment

The authors express their gratitude to the Thiagarajar College of Engineering (TCE) for supporting us to carry out this research work.

References

[1]Nojiri, N., Kong, X., Meng, L., & Shimakawa, H. (2019). Discussion on machine learning and deep learning based makeup considered eye status recognition for driver drowsiness. *Procedia computer science*, *147*, 264-270.

[2]Fuletra, J. D., & Bosamiya, D. (2013). A survey on drivers drowsiness detection techniques. *International Journal on Recent and Innovation Trends in Computing and Communication*, *1*(11), 816-819.

[3]Ramos, A. L. A., Erandio, J. C., Mangilaya, D. H. T., Carmen, N. D., Enteria, E. M., & Enriquez, L. J. (2019). Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis. *International Journal of Simulation--Systems, Science & Technology*, *20*.

[4]Saini, V., & Saini, R. (2014). Driver drowsiness detection system and techniques: a review. *International Journal of Computer Science and Information Technologies*, *5*(3), 4245-4249.

[5]Ghimire, D., Jeong, S., Yoon, S., Park, S., & Choi, J. (2015). Real-time sleepiness detection for driver state monitoring system. *Advanced Science and Technology Letters*, *120*, 1-8.

[6]Deng, W., & Wu, R. (2019). Real-time driver-drowsiness detection system using facial features. *Ieee Access*, *7*, 118727-118738.

[7]Poursadeghiyan, M., Mazloumi, A., Saraji, G. N., Baneshi, M. M., Khammar, A., & Ebrahimi, M. H. (2018). Using image processing in the proposed drowsiness detection system design. *Iranian journal of public health*, *47*(9), 1371.

[8]Ngxande, M., Tapamo, J. R., & Burke, M. (2017). Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques. *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 156-161.

[9]Jin, L., Niu, Q., Jiang, Y., Xian, H., Qin, Y., & Xu, M. (2013). Driver sleepiness detection system based on eye movements variables. *Advances in Mechanical Engineering*, *5*, 648431.

[10]Clement, F. S. C., Vashistha, A., & Rane, M. E. (2015, December). Driver fatigue detection system. In *2015 International Conference on Information Processing (ICIP)* (pp. 229-234). IEEE.

[11]He, J., Roberson, S., Fields, B., Peng, J., Cielocha, S., & Coltea, J. (2013). Fatigue detection using smartphones. *Journal of Ergonomics*, *3*(03), 1-7.

[12]Joshi, K. V., Kangda, A., & Patel, S. (2016). Real time system for student fatigue detection during online learning. *Int J Hybrid Inf Technol*, *9*(3), 341-346.