Journal of Engineering Research

Volume 6 | Issue 3 Article 24

2022

A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction

Mohamed Gomaa

A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction

Mohamed W. Gomaa [b]*, **Rasha O. Mahmoud** [b], **Amany M. Sarhan** [a]

a Department of Computer and Control Engineering, Faculty of Engineering, Tanta University, Tanta 37133, Egypt.
b Faculty of Computer and Artificial Intelligence, Benha University, Benha, Egypt
* Corresponding Author Email Addresses: mohamed.waheed16@fci.bu.edu.eg

Abstract: **The development of neural networks and machine learning techniques has recently been the cornerstone for many applications of artificial intelligence. These applications are now found in practically all aspects of our daily life. Predicting drowsiness is one of the most particularly valuable of artificial intelligence for reducing the rate of traffic accidents. According to earlier studies, drowsy driving is at responsible for 25 to 50% of all traffic accidents, which account for 1,200 deaths and 76,000 injuries annually. The goal of this research is to diminish car accidents caused by drowsy drivers. This research tests a number of popular deep learning-based models and presents a novel deep learning-based model for predicting driver drowsiness using a combination of convolutional neural networks (CNN) and Long-Short-Term Memory (LSTM) to achieve results that are superior to those of state-of-the-art methods. Utilizing convolutional layers, CNN has excellent feature extraction abilities, whereas LSTM can learn sequential dependencies. The National Tsing Hua University (NTHU) driver drowsiness dataset is used to test the model and compare it to several other current models as well as state-of-the-art models. The proposed model outperformed state-of-the-art models, with results up to 98.30% for training accuracy and 97.31% for validation accuracy.**

Keywords— **Driver Drowsiness, Deep learning, CNN, LSTM, NTHU dataset.**

1. INTRODUCTION

It has been proven that a human should sleep up to 8 hours a day to improve his fitness. Since sleep activity is not a choice, it is essential to continue working and focusing on the other activities. Drowsiness was noticed to occur in two times through a day; in the early morning and late in the afternoon. Furthermore, the frequent and recurring disorder of human sleep causes drowsiness during normal life activities during the day [18, 35, 38]. Sleep deficiency, moreover, harms human health and gradually and cumulatively leads to severe conditions. Nevertheless, the only effective way to diminish sleepiness is to have a sleeping time where the body gets its required rest and can continue its proper functions. According to a study of road accidents in Germany, an estimated 35% of fatal road accidents are due to attention disorders caused by drivers in fatigue condition. In New Zealand, between 1996 and 1998, there were 114 traffic accidents and 1,314 traffic accidents with injuries related to driver drowsiness. A 1997 study of 370 major auto accidents indicated that driver drowsiness was a contributing factor in 7% of all accidents [18, 38].

Recent statistics estimate that annually 1,200 deaths and 76,000 injuries are due to driver's exhaustion or drowsiness crashes [33]. According to the National Highway Traffic Safety Administration (NHTSA), 4% of drivers fall asleep during the month. The National Highway Traffic Safety Administration (NHTSA) reports that sleep deprivation causes nearly 100,000 crashes each year in the United States. However, this figure is an underestimated one. According to the British Center for Sleep Research, driver exhaustion accounts for 20% of all road accidents. The Australian Road Safety Agency estimates that 25-35% (up to 50 per cent) of road accidents are associated with excessive sleep. A study of 9200 crashes in Norway found that 4% had sleep-related crashes and 20% night-time crashes that left drivers asleep. According to the World Health Organization, road accidents caused the deaths of around 1.25 million people in 2015. This means that fatal accidents occur about every 25 seconds.

To reduce and eliminate drowsiness-related road accidents, it is necessary to construct a highly- efficient system to predict driver drowsiness. One of the directions to achieve this goal is the development of an altering system that monitors the drivers while they are driving and detects drowsiness condition. However, creating such a system is not an easy task and requires intelligence. There has been a significant effort in the literature to develop accurate drowsiness detection systems. All these systems are based on a machine learning model, as will be explained in the related work section.

Previous work directed towards driver drowsiness detection had used SVM [29, 32], ML classified algorithms [37], CNN [8-11, 19, 30, 36, 39], Image processing [27], lightweight CNNs [20], and [28]), pre-trained CNN [2], LSTM model [7], ML algorithms (HOG) and (NB) [4], CNN and face alignment [41], and CNN combined with RCNN [12]. While the highest performance achieved in the previous work on the Brain4cars dataset [22] was 96.05 % using CNN combined with RCNN [12]. Other datasets are also available. However, those are the two most famous datasets. Nevertheless, the accuracy reached by most of the previous work is not sufficient to rely on when building real-time systems. Therefore, more robust and accurate solutions should be introduced.

In this paper, we introduce a drowsy driver prediction approach using deep learning models in order to enhance its prediction accuracy. In order to predict drivers' tiredness, we trained and evaluated a collection of well-known CNN architectures, including ResNet, Vgg-16, GoogleNet, and MobileNet. However, in the case at hand in this research, these models fall far short of the level of accuracy that was expected of them. We build our combined CNN-LSTM model as a response to enable maximum prediction accuracy outcomes.

One of the most well-known neural network models to be extensively applied in computer vision applications is CNN [40]. Several building blocks, including convolution layers, pooling, and fully connected, are used to adapt and automatically learn spatial hierarchies of features over back-propagation. On the other hand, LSTM is an artificial recurrent neural network (RNN) that has feedback connections and can pass single data points like images or sequences of data like videos. In earlier research, combining CNN and LSTM models generated better results; this model will integrate the advantages of both models. The CNN model is chosen because, compared to other deep learning models, it can handle vast volumes of raw data with comparatively little preprocessing work [2]. This makes it easier to use the model in real-time applications.

Utilizing LSTM can improve back propagation optimization. As a result, the combination of CNN and LSTM can enhance system accuracy while also offering optimal real-time system utilization. When trained and evaluated using the National Tsing Hua University (NTHU) driver drowsiness dataset (introduced by the Computer Vision Lab, National Tsuing Hua University) [5], the proposed CNN-LSTM drivers' drowsiness prediction system demonstrated to outperform the relevant state-of-the-art systems in efficiency.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of earlier research that is relevant to the topic of the paper. In Section 3, the proposed system is described. The experimental findings of the proposed model are presented in section 4. Section 5 presents the conclusion and possible future work.

2. RELATED WORK

Deep learning is one of the modern and advanced artificial intelligence methods that help computers learn, which is what people easily excel at. For example, through deep learning, it is possible to manage self-driving vehicles, which can perceive the presence of obstacles in the road or recognize the presence of pedestrians and thus make the right decision. Currently, we have many applications of deep learning, such as voice control of phones and tablets, or the recognition of people under different conditions. Deep learning achieves unexpected results in previous methods and achieves high accuracy, exceeding humans, especially in response speed and accuracy. Deep learning models are built using neural network structures that have many layers and are higher than traditional learning methods, so they are called "deep neural networks". Figure 1 shows the general architecture of the neural network that is used to build deep neural networks (DNN).

One of the deep learning algorithms, and the first introduced, is the convolutional neural network (CNN), which can take an input image, assign learnable task weights to different objects in the image, and be able to distinguish one from the other. CNN requires less configuration and preprocessing than other machine learning (ML) algorithms. Although the filters are mainly used manually, with proper training, CNN can recognize these filters. A simple CNN example is shown in Figure 2.

The pooling layer is one of the main concepts of CNN. It reduces the computational burden by decreasing the number of connections between convolutional layers, and the assembly can generally be represented by [9]:

$$y\_(i,j,k) = [\sum_{(m.n)\in R(ij)} (A_{m,n,k}{}^{p}]^{\frac{1}{p}} \qquad (1)$$

Relu or "Rectified linear unit" is one of the most notable non-saturated activation functions. The Relu activation function is [9]:

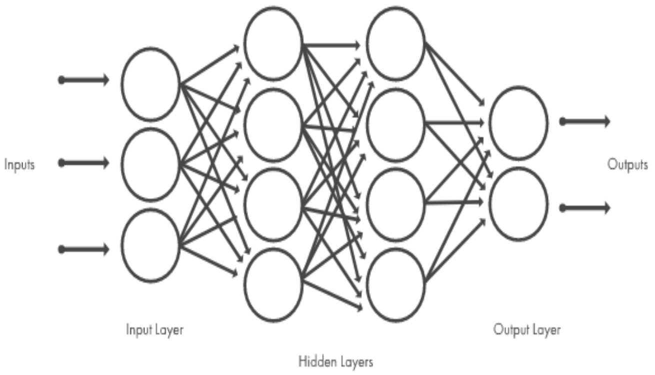


Figure 1. Neural network structure.

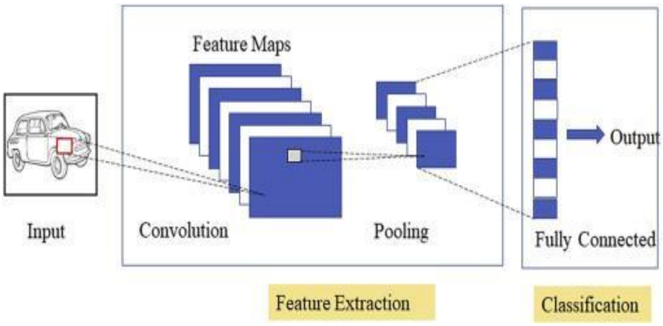


Figure 2. Convolutional Neural Network (CNN).

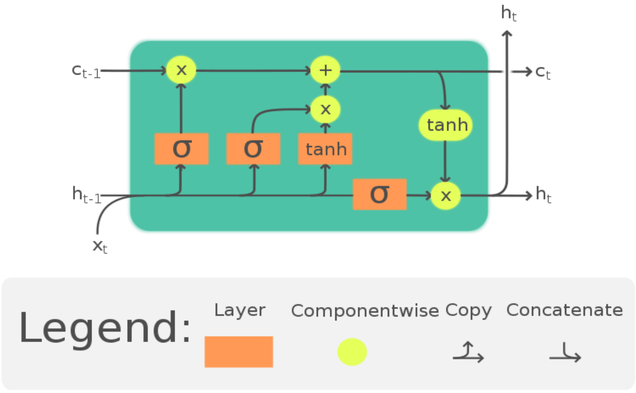


Figure 3. LSTM cells that process data sequentially and keep their hidden state through time.

$$a_{(i,j,k)} = \max(Z_{i,j,k}, 0) \qquad (2)$$

A model is said to be overfit when it learns statistical regularities specific to the training set, memorizing the irrelevant noise rather than the signal, and then performs poorly on a fresh dataset after that. One of the fundamental issues with machine learning is that an overfitted model cannot be applied to data that has never been seen before. As was mentioned in the section above, a test set is crucial in this respect for properly evaluating the performance of machine learning models [40].

Recurrent neural networks (RNN) are considered to have been improved by long short-term memory (LSTM) [7]. In order to solve the issue of vanishing and exploding gradient, memory blocks are used instead of traditional RNN units. It augments a cell state to store the long-term states, which is the key difference introduced by RNNs. The LSTM network can, therefore, remember the previous information and deliver it to the current data. A simple figure to represent the idea of the LSTM cell is shown in Figure 3.

To reduce the number of drowsiness-related accidents, there is a persistent need to predict driver drowsiness. Deep learning is one of the most effective techniques that is used to identify driver drowsiness with the highest accuracy. Many recent approaches are applied to the prediction of driver drowsiness. We will briefly mention some of them.

We start by listing the early versions of driver drowsiness systems which were based on machine learning approaches. An SVM model was introduced to detect driver drowsiness based on tracking mouth for yawning recognition [29]. During the detection process, the mouth area is detected from face images using a series of classifiers. The SVM model was then used to classify the mouth status and to detect yawning alerting a fatigue condition. They used 20 yawning images and more than 1000 normal images. Their results show an accuracy of 86% for normal images and 81% for yawning images.

Several machine learning approaches were employed to determine actual human behavior during drowsiness episodes [37]. They used machine learning-based classifiers (algorithms) such as Adaboost and multinomial ridge regression on a separate database of natural expressions containing; facial blinking, yawning motions, and other facial movements. They used two datasets: Cohn-Kanade DFAT-504, and directed facial actions from 24 subjects. The training set contained 6,000 images. Their results show an accuracy of 90% across all subjects.

An image-processing technique and Viola-Jones face detection algorithms were proposed by Poursadeghiyan et al. [27]. In their framework, they aim to detect levels of drowsiness. They used eye-tracking techniques including eye blink duration, blink frequency, and PERCLOS. A driving simulator was used for testing. Their results show an accuracy of 93%.

Subbaiah et al. [32] applied SVM for drowsiness detection. In their model, the best features were chosen from input videos; like eyes, mouth region, and head pose angle. They applied the Infinite Feature Selection (IFS) algorithm to select the optimized features. Such features were fed to the SVM binary classification algorithm to classify the stages of drowsiness detection. NTHU is used to test the model performance. The model accuracy was 88.97%.

Bakheet & Al-Hamadi [4] introduced a machine learning approach for driver drowsiness detection. It relied on Histogram of Oriented Gradients (HOG) for features extraction, object detection, and Naïve Bayes (NB). They applied their approach to the NTHU video dataset, and they obtained 85.62% accuracy.

Other research work relied on deep learning approaches for the prediction task, such as [10], who employed a CNN model to detect driver drowsiness. The model was based on blink rate, eye closure, yawning, eyebrow shape, and other hand-engineered facial features. A SoftMax layer was used to classify the driver as drowsy or non-drowsy, using a dataset collected from 30 different subjects in diverse conditions. Their results show an accuracy of 92.33%.

Another CNN model was introduced by Jabbar et al. [19] to be implemented as a real-time Android application, where a minimal network structure was designed. It was based on facial landmark key point detection to recognize whether the driver was drowsy or not. They used the NTHU video dataset; which contains 22 subjects from different ethnicities and 9.5 hours of videos with 640*480 resolution at 15-30 fps containing different subjects enacting regular and drowsy driving behaviors. The accuracy level reached was more than 80%.

A CNN model for real-time low-cost embedded systems was proposed by Reddy et al. [28]. Their model aims to be a compression of a heavy baseline model to a lightweight model deployable on an embedded board. They used a minimized network structure based on facial landmarks as input to recognize whether the driver was drowsy or not. They used a custom-collected dataset. In their model, the accuracy average was 89.5%.

The MobileNet CNN model - originally proposed by Howard et al. [16] - was used for drowsiness detection from a video stream of a driver by Shakeel et al. [30]. It detected and localized open and closed eyes along with the Single Shot Multibox Detector (SSD) [24], and (MobileNet-SSD) [17] for object detection and drowsiness detection. They built an augmented dataset from both “FDDB” (Face Detection Dataset and Benchmark [21]) containing 2845 images and 5171 faces, Yawing dataset (YAWDD) [1], and closed eyes in the wild (CEW) by Song et al. [31]. Their model accuracy was 83.7%.

Another CNN model was proposed by de Nauroisa et al. [8] to detect drowsiness. In addition, the model aims to predict when a given drowsiness level is reached. The model used eyelid movements and recorded driving behaviors; such as time-to-lane-crossing, speed, steering wheel angle, and position in the lane. They used a special dataset containing videos of 21 participants who drove a car simulator for 110 minutes under optimized conditions to induce drowsiness. Their model accuracy was 95%.

A CNN model was proposed by Wijnands et al. [39], which aimed to build real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks. The model was pre-trained using ImageNet and Kinetics where the NTHU dataset was used. The model's accuracy was 82%. Jabbar et al. [20] proposed a lightweight CNN model to detect driver drowsiness Heavier classification categories were used (i.e., many eye situations like with/ without glasses, and day/night imaging). The NTHU dataset was used, and the average accuracy was more than 83% for all categories.

Abbas [2] proposed a pre-trained model (CNN) based on eye recognition, yawn analysis, and classification of electrocardiography (ECG) signals. They used ECG signals for driver Heart Rate Variability (HRV) measurement during highway driving under different conditions. Three datasets were used: the Closed Eyes in the Wild (CEW) dataset, the Yawing dataset (YAWDD) [1], and the Columbia Gaze dataset (CAVE-DB). They obtained 93% average accuracy in real-time testing.

An LSTM model for driver drowsiness prediction was proposed by Chui et al. [7]. Empirical Mode Decomposition (EMD) was applied to ECG samples to obtain the sum of all intrinsic mode functions (IMFs), and then they were fed to the LSTM model. The system was tested using two datasets; Cyclic Alternating Pattern Sleep (CAPS) [34] and Stress Recognition in Automobile Drivers (SRAD) [13]. The accuracy results obtained were 72.2% for the CAPS dataset and 81.5% for the SRAD dataset.

Another CNN model for driver drowsiness detection was proposed by Dua et al. [9]. The model relied on AlexNet, VGG-FaceNet, FlowImageNet, and ResNet. The model extracted environmental features from AlexNet, facial features from VGG-FaceNet, behavioral features from FlowImageNet, and yawning with hand gestures features from ResNet. The results were assembled manually, and they used the NTHU video dataset. Their model accuracy results are: AlexNet 76.48%, VGG-FaceNet 87.09%, FlowImageNet 83.11%, ResNet 81.34%, and Ensemble 85%.

Faraji et al. [11] proposed a hybrid CNN and LSTM model to detect and predict driver drowsiness. In their model, YOLOv3 CNN was used for object detection like closed eyes, face, mouth, and yawn. The output of the CNN was fed as an input to RNN with LSTM. For the framework performance, they applied the model to a dataset consisting of 28 RGB videos extracted into 1048 images. They obtained 91.7% accuracy.

A CNN model was proposed in [36] to detect driver drowsiness. The authors extracted eyes from each frame using dlib’s API, and then fed it to the CNN eye classification model to predict eye state as open or closed. The eye state was stored and analyzed to predict if the driver was drowsy or not. For testing the performance of their framework, they applied it to the MRL Eye dataset [25]; which consists of 84k human eye images captured in various driving conditions. They obtained a training accuracy of 95%, and a validation accuracy of 90%.

A CNN equipped with face alignment model was introduced by Xiaofeng et al. [41]. Their model aimed to build a driver fatigue detection model. Face detection network LittleFace is used with a speed optimized SDM algorithm, then its output is the input of the CNN model to detect driver fatigue. The Yawing Detection Dataset (YAWDD) was used to test the model's performance [1]. The dataset contains 322 videos recorded by an in-car camera of drivers with various facial characteristics (male/female, with/without glasses, or sunglasses, talking, singing, being silent, and yawning). The accuracy of their model was 89.55%. Table 1 provides the summary of the abovementioned techniques in this section.

An Advanced Driver Movement Tracking (ADMT) system was created using recurrent neural network (RCNN) and CNN, a contemporary deep learning technique [12]. Comparatively speaking, that system outperformed SVM, FFNN, and F-RNN-EL. The model was trained and tested using the Brain4cars dataset for driver movement tracking, introduced by A. Jain et al. [22], and obtained 96.05 percent performance accuracy. The Mobile-Nets Model and LSTM have been integrated on a well-known benchmark dataset to create another deep learning model for the detection of driver drowsiness [3]. Despite only achieving performance accuracy of 80%, they proved that the model could be used with a frame rate of 80% on a commercial and affordable development board with a frame rate of 5 frames per second.

The aforementioned approaches' inadequate accuracy (varying from 76.48 % to 96.05 %) necessitates the development of new ways that aid in boosting the detection systems' accuracy. In order to improve the performance of the current systems in terms of prediction accuracy, we offer a number of deep learning prediction models for the driver drowsiness problem.

Table 1. Summarization of the Related Work for drowsiness detection

Paper	Year	Method	Dataset	Accuracy
Saradadevi & Bajaj [29]	2008	SVM was used to classify the mouth, detect yawning, and alert Fatigue by monitoring and recognizing yawning.	1000 normal images. 20 yawning images.	Normal images: 86% Yawning image: 81%.
Vural et al. [37]	2008	Machine learning-based classifiers (algorithms) such as Adaboost and multinomial ridge regression on a separate database of spontaneous expressions.	Cohn-Kanade DFAT-504 dataset. Total training set consisted of 6000 examples.	90% accuracy across all subjects.
Dwivedi et al. [10]	2014	CNN model based on blink rate, eye closure, yawning, eyebrow shape, and other hand-engineered facial features. A SoftMax layer is used to classify the driver as drowsy or non-drowsy.	Collected from 30 different subjects in diverse conditions.	92.33%
Poursadeghiyan et al. [27]	2018	Image-processing technique and Viola-Jones face detection algorithms for detecting levels of drowsiness. They used eye-tracking including eye blink duration, blink frequency, and PERCLOS. A driving simulator was used for testing.	Training: 9964 frames recorded from 5 drivers. Testing: driving simulator.	93%.
Jabbar et al. [19]	2018	Real-time CNNs model as Android application. Facial landmark key point to recognize whether the driver was drowsy.	NTHU video dataset [5]	More than 80%.
Reddy et al. [28]	2017	Lightweight CNNs model for a real-time low-cost embedded system to use inside the car. Facial landmark input to recognize whether the driver was drowsy or not.	Custom collected dataset.	89.5%
Shakeel et al. [30]	2019	CNN model for drowsiness detection from the video stream of a driver. It detects and localizes open and closed eyes. MobileNet CNN architecture along with the Single Shot Multibox Detector (SSD), (MobileNet-SSD) for object detection and drowsiness detection.	Combined dataset from: “FDDB” (Face Detection Data Set and Benchmark) [21], and Yawing dataset (YAWDD) [1], Closed Eyes in the wild (CEW) [31].	83.7%
de Nauroisa et al. [8]	2019	CNN model to detect drowsiness and predict when a given drowsiness level reached as well. They used eyelid movements and recorded driving behavior such as time-to-lane-crossing, speed, steering wheel angle, and position on the lane.	A special dataset contains 21 participants who drove a car simulator for 110 min under optimized conditions to induce drowsiness.	95%

Wijnands et al. [39]	2020	CNN model to build real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks The model was pre-trained on ImageNet and Kinetics.	NTHU video dataset.	82%
Jabbar et al. [20]	2020	A lightweight CNN model was used as a classification model. Eye categories; with/without glasses, and day/night.	NTHU video dataset.	Average more than 83% in all categories.
Abbas [2]	2020	Pre-trained (CNN) model based on eye recognition, yawn analysis, and classification of (ECG) signals. ECG signals for driver Heart Rate Variability (HRV) measurement during highway driving under different conditions.	Closed eyes in the wild (CEW) dataset, Yawing dataset (YAWDD) [1], Columbia gaze dataset (CAVE-DB).	93% average accuracy on the real-time test dataset.
Subbaiah et al. [32]	2020	SVM was used for drowsiness detection. The best features were chosen from input videos; like eyes, mouth region, and head pose angle. Infinite Feature Selection (IFS) algorithm was applied to select the optimized features. Such features were fed to the SVM binary classification algorithm to classify the stages of drowsiness detection.	NTHU video dataset.	88.97%
Chui et al. [7]	2021	An LSTM model for driver drowsiness prediction. Empirical Mode Decomposition (EMD) was applied to ECG samples to obtain the sum of all intrinsic mode functions (IMFs), then they are fed to the LSTM model.	Cyclic Alternating Pattern Sleep (CAPS) – [34]. Stress Recognition in Automobile Drivers (SRAD) [13].	CAPS: 72.2%. SRAD: 81.5%.
Dua et al. [9]	2021	CNN-based models: AlexNet, VGG- FaceNet, FlowImageNet, and ResNet for driver drowsiness detection. Extracting environmental features from AlexNet, facial features from VGG- FaceNet, behavioral features from FlowImageNet, and yawning with hand-gesture features from ResNet and assembling the results manually.	NTHU video dataset.	AlexNet: 76.48% VGG-FaceNet: 87.09% FlowImageNet: 83.11% ResNet: 81.34% Ensemble: 85%
Bakheet & Al-Hamadi [4]	2021	Histogram of Oriented Gradients (HOG) for features extraction and object detection Naïve Bayes (NB) for Driver Drowsiness Detection.	NTHU video dataset.	85.62%
Faraji et al. [11]	2021	Hybrid CNN and LSTM in parallel for detection and prediction of driver drowsiness. YOLOv3 CNN was used for object detection like closed eyes, face, mouse, and yawning. CNN output is the input to recurrent neural networks RNN with LSTM	28 RGB videos extracted to 1048 images.	91.7%
Tibrewal et al. [36]	2021	CNN to detect driver drowsiness. Eyes were extracted from each frame using dlib's API and then fed to the CNN eye classification model to predict eye state as open or closed. The eye state was stored and analyzed to predict if the driver was drowsy or not.	MRL Eye Dataset [25] of 84k human eye images captured in various driving conditions.	95% training accuracy, 90% validation accuracy.
Xiaofeng et al. [41]	2021	CNN model and face alignment to build driver fatigue detection model. Face detection network LittleFace was used with a speed-optimized SDM algorithm, then input it to the CNN model to detect driver fatigue.	YAWDD: yawing detection dataset [1], contains 322 videos recorded by an in-car camera, of drivers in an actual car with various facial characteristics.	89.55%
Gite et al. [12]	2021	CNN combined with RCNN was used to build an Advanced Driver Movement Tracking (ADMT) system.	Brain4cars dataset for driver movement tracking [22]	96.05 %.
Aydemir et al. [3]	2021	Mobile Nets Model has combined with Long Short-term Memory.	Brain4cars dataset for driver movement tracking [22]	80%.

3. PROPOSED SYSTEM

In this paper, a deep learning classification model is introduced to predict the driver's drowsiness from a video captured while he is driving. Several models will be tested and evaluated to achieve the best possible accuracy. In the proposed model, Google "Colab Pro Plus version" has been used as the execution platform. Additionally, the models were developed, trained, and evaluated using Python-based TensorFlow and Keras [14].

3.1. Dataset for the study

In our research, we used the National Tsing Hua University (NTHU) Driver Drowsiness Dataset, which was developed by Ching-Hua et al. There are 22 distinct subjects represented in it (male and female). Every video could have both drowsy and non-drowsy statuses. Every video features a unique circumstance with several state changes. Videos were recorded with and without wearing glasses/sunglasses under a variety of simulated driving scenarios, including normal driving, yawning, slow blink rate, falling asleep, burst out laughing… etc., under day and night illumination conditions.

A driving simulator with wheels and pedals was used to record the videos. The overall dataset size was roughly 6.5 GB, and the total dataset time was about 9.5 hours. They acquired IR videos using active infrared (IR) light to gather the dataset. The videos had an average frame rate of 15 to 30 frames, and their resolution was 640x480 with AVI-format files. Examples of photos from the NTHU dataset are shown in Figures 4 and 5.

As illustrated in Figures 6, 7, and 8, the following operations were used as data augmentation techniques to expand the dataset size:

- Rotation: samples have been rotated at 90, 180 degrees.
- Flipping: samples were up-down flipped.
- Mirroring: samples were right/left mirrored.

3.2. Dataset preprocessing

The dataset includes a label file for each video. The videos need to be preprocessed in the proposed model in order to use them. As demonstrated in Figure 9, a TensorFlow code [14] is first applied to the dataset videos for validation in order to display the label value on the video. All dataset labels have been changed by time-shifting drowsy values to an earlier time after evaluating the dataset videos so that the suggested model can make predictions. TensorFlow [14] created a generator that loops over all movies, separates them into frames, and resizes each frame to be 112x112 in size. The model receives frames with the appropriate labels from the generator. An illustration of video conversion to frames is shown in Figure 10.

The proposed model was optimized using Keras during the training phase by using a Call-back function. Prior to overfitting, training has been halted using the Early-Stop Keras function. The best model weight has been saved using the Checkpoint (Keras function) in the Call-back function during training. The ReduceLROnPlateau Keras function was employed during the learning process to lower the learning rate if the learning accuracy was not improving.

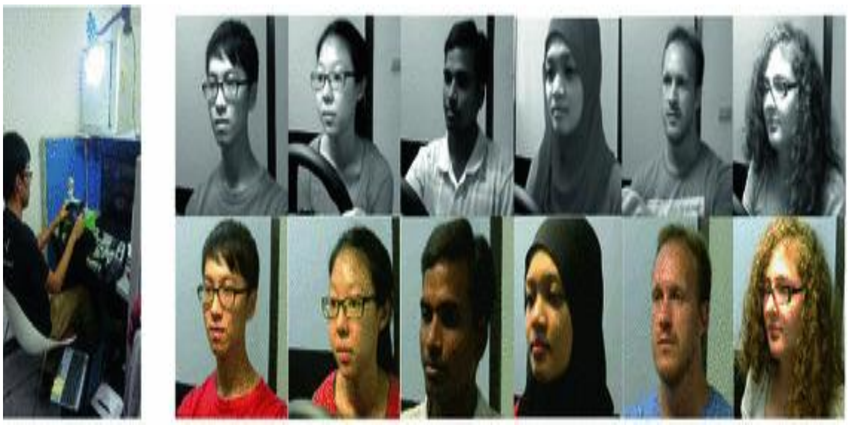


Figure 4. Samples of participants - NTHU dataset [5]

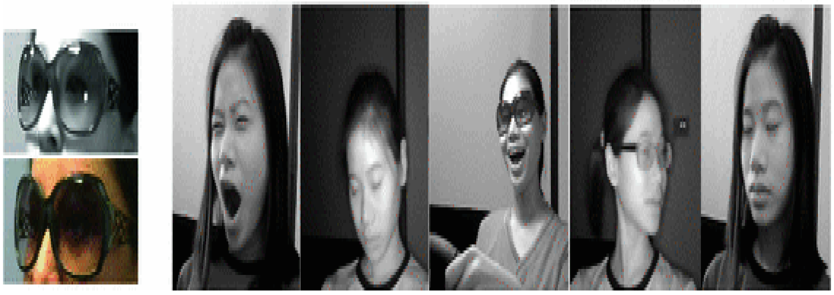


Figure 5. Different scenarios and performances (NTHU dataset) [5] **showing different illumination and pose conditions, and wearing glasses.**

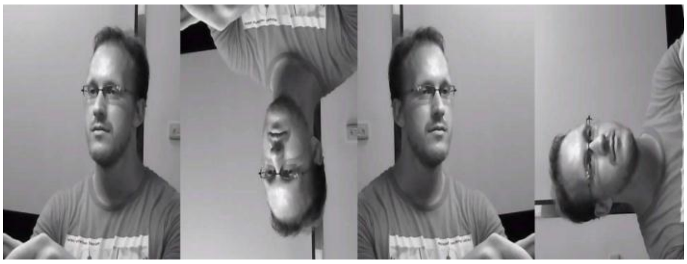


Figure 6. Data augmentation using rotation of images.

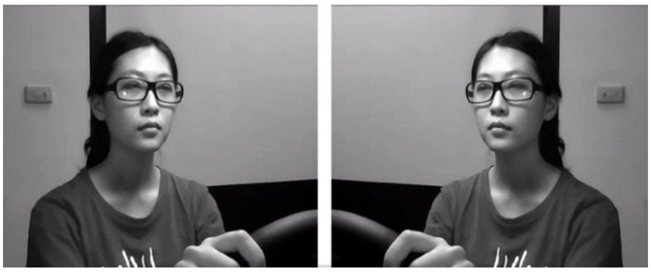


Figure 7. Data augmentation using mirroring of images.

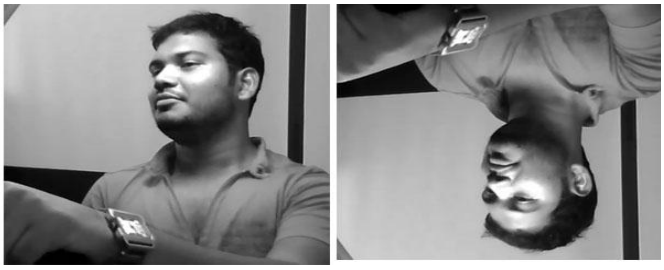


Figure 8. Data augmentation using flipping of images.

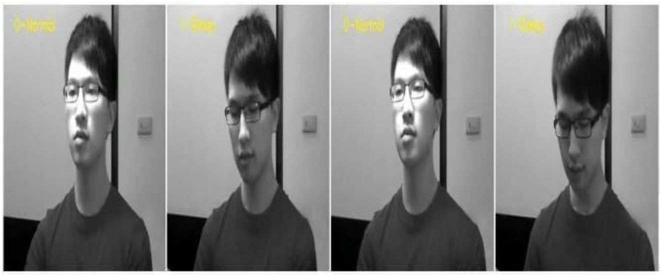


Figure 9. Dataset validation of labels (Drowsy or Not).

3.3 Tested models

In an experiment to determine the optimal model for the system, a number of CNN-based models, including ResNet, VGG-16, GoogleNet, and MobileNet, were utilized to develop the driver's drowsiness model. The specifics of their architecture will be introduced in the subsections that follow.

3.3.1. ResNet model

The ResNet model is one of the most famous deep learning models. Its architecture consists of 152 deep CNN layers, Conv 64(7x7), 3 Conv 64(1x1), 64(3x3), 256(1x1), then 3 Conv 128(1x1), 128(3x3), 512(1x1), followed by 3 Conv 256(1x1), 256(3x3), 1024(1x1) and 3 Conv 512(1x1), 512(3x3), 2048(1x1), then a fully connected layer as shown in Figure 11.

3.3.2. VGG-16 model

The VGG model is one of the standard deep learning models [23]. Its architecture consists of 16 layers: 2 Conv (224x224x64) with max-pooling layer, 2 Conv (112x112x128) with max-pooling layer, then 3 Conv2D (256) with max-pooling layer, 3 Conv2D (512) with max-pooling layer, 3 Conv2D (512) with max-pooling layer then fully connected layers as shown in Figure 12.

3.3.3 GoogleNet model

Google-Net model, which is a convolutional neural network that consists of 22 layers: 9 inception modules, has been also tried. Each inception module ends with a global average pooling [15] as shown in Figure 13.

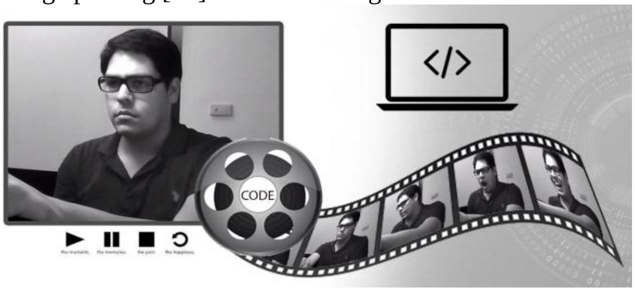


Figure 10. Converting Video into Frames.

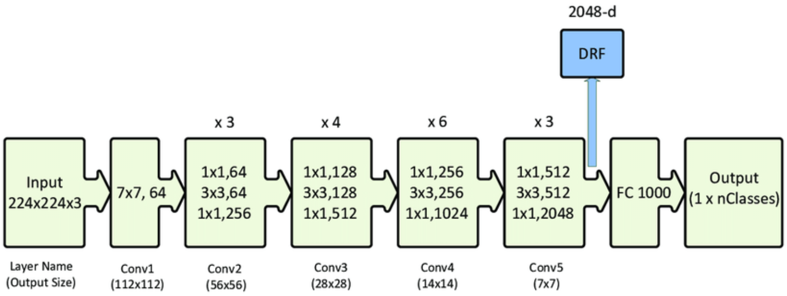


Figure 11. ResNet deep learning model architecture.

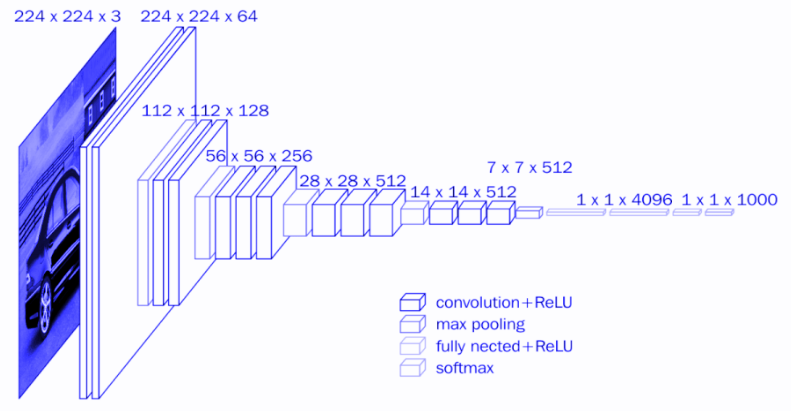


Figure 12. VGG-16 deep learning model architecture [23]

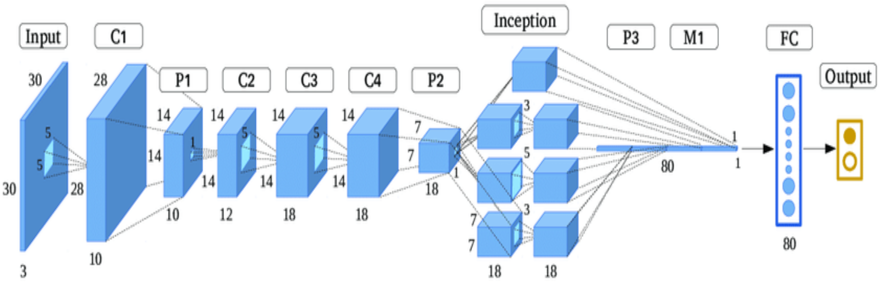


Figure 13. GoogleNet deep learning model architecture [15].

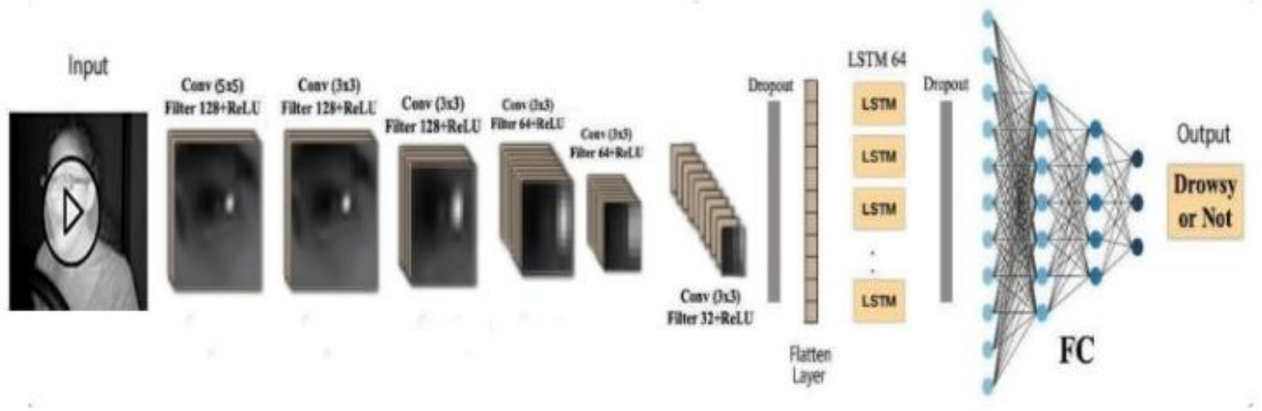
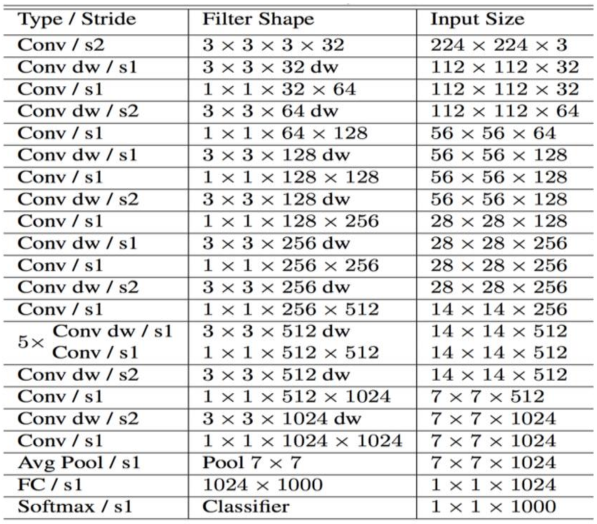

Input
Conv (5x5) Filter 128+ReLU
Conv (3x3) Filter 128+ReLU
Conv (3x3) Filter 128+ReLU
Conv (3x3) Filter 64+ReLU
Conv (3x3) Filter 64+ReLU
Conv (3x3) Filter 32+ReLU
Dropout
Flatten Layer
LSTM 64
LSTM
LSTM
LSTM
LSTM
Dropout
FC
Output
Drowsy or Not

Figure 15. The proposed (CNN-LSTM) model architecture.

3.3.4 MobileNet model

MobileNet is a deep learning lightweight CNN model which is commonly used for faster training, mobile apps, and embedded systems [26]. MobileNet consists of 28 layers as shown in Figure 14.



Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 14. MobileNet architecture and parameters [26]

3.4. The proposed driver drowsiness detection system based on (CNN-LSTM) model

In this research, we provide a driver drowsiness detection system that is built on a composite deep learning model that combines two well-known powerful models: CNN and LSTM. Figure 15 depicts the overall system design. The combination of CNN and LSTM models has been demonstrated in numerous applications to significantly improve the classification of the data and to capture the dynamic information concealed in the video sequence.

The CNN layers are successful in extracting features from the individual video frames. In order to comprehend the dynamic features in each frame of the sequence, the LSTM layer needs this information. By specifying the complex temporal dynamics of the video, this interpretation allows the entire model to capture them, improving the classification accuracy by defining the perceptual representation of frames in the convolutions well.

The proposed CNN-LSTM deep learning model is first trained using the videos included in the dataset before being applied to identify the driver's level of drowsiness. Figure 16 depicts the proposed CNN-LSTM deep learning model, which is formed of a CNN network followed by an LSTM network.

The LSTM is used in order to utilize the memory cells to store, modify, and access the internal state and hence discover the temporal information contained in the input video. The CNN network's output is refined and forwarded through time and upwards through the layers of LSTM. A fully connected layer is then utlized to classify the output.

The proposed driver drowsiness prediction model (CNN-LSTM) consists of ConV (5x5) with filter 128, an activation function "Relu" with a MaxPooling layer, a batch normalization layer, ConV (3x3) with filter 128, an activation function "Relu", a MaxPooling layer, a Batch Normalization Layer, ConV (3x3) with filter 64, an activation function "Relu", a MaxPooling layer, a Batch Normalization Layer, ConV (3x3) with filter 64, an activation function "Relu", a MaxPooling layer, a batch normalization layer, ConV (3x3), filter 32 with an activation function "Relu", a MaxPooling layer, a batch normalization layer then a flatten layer.

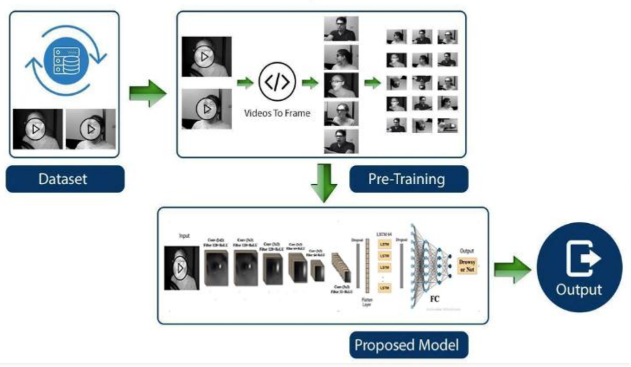


Figure 16. The proposed CNN-LSTM model training/testing process description.

It has been used as input to LSTM (64) with Dropout 0.4, then the output of LSTM was fed to fully connected layers as Dense 128 with activation function "Relu" with Dropout 0.5, Dense 64 with activation function "Relu" with Dropout 0.4, Dense 32 with activation function "Relu" with Dropout 0.4, and Dense 16 with activation function "Relu" with Dropout 0.2. A sigmoid layer has also been used for the final output. The total number of parameters of the CNN-LSTM model is

397,729, while the trainable parameters are 396,897. The details of the proposed model architecture and parameters are given in Table 2.

4. EXPERIMENTAL RESULTS

About 9.5 hours of videos are contained in the dataset, which was provided by the computer vision lab at National Tsuing Hua University [5]. The videos are then transformed to frames using TensorFlow. Then, using a cross-validation strategy, they were divided into two sets: 20% were kept as a testing set and 80% were used as a training set. The Keras "Early-Stop" function and smart learning rate function "ReduceLROnPlateau" have been employed in each epoch of the SGD optimizer, starting with 1e-3. If accuracy is not maximized throughout the training process, the learning rate is reduced after some epochs. In both CNN and LSTM, the batch size is 16, and the number of epochs is 45, respectively. The network employs a number of dropout layers and batch normalization to avoid overfitting. The model accuracy and loss curves are shown in Figures 20 and 21.

On the same dataset, the accuracy of the proposed model is first compared with that of the well-known CNN-based models. The accuracy of the proposed combined network, which combines the CNN and LSTM, is then evaluated in comparison to recent earlier work, which used the same dataset and produced the best results to date.

Table 2. The proposed model architecture details and parameters

Layer (type)	Act.	Output Shape	Param.
TimeDistribution(ConV)	Relu	(None, 16, 108, 108, 128)	9728
TD(MaxPooling)		(None, 16, 54, 54, 128)	0
batch_normalization		(None, 16, 54, 54, 128)	512
TD(ConV)	Relu	(None, 16, 52, 52, 128)	147584
TD(MaxPooling)		(None, 16, 26, 26, 128)	0
batch_normalization		(None, 16, 26, 26, 128)	512
TD(ConV)	Relu	(None, 16, 24, 24, 64)	73792
TD(MaxPooling)		(None, 16, 12, 12, 64)	0
batch_normalization		(None, 16, 12, 12, 64)	256
TD(ConV)	Relu	(None, 16, 10, 10, 64)	36928
TD(MaxPooling)		(None, 16, 5, 5, 64)	0
batch_normalization		(None, 16, 5, 5, 64)	256
TD(ConV)	Relu	(None, 16, 5, 5, 32)	18464
TD(MaxPooling)		(None, 16, 3, 3, 32)	0
batch_normalization		(None, 16, 3, 3, 32)	128
TD(Flatten)		(None, 16, 288)	0
lstm (LSTM)		(None, 64)	90368
Dropout		(None, 64)	0
Dense	Relu	(None, 128)	8320
Dropout		(None, 128)	0
Dense	Relu	(None, 64)	8256
Dropout		(None, 64)	0
Dense	Relu	(None, 32)	2080
Dropout		(None, 32)	0
Dense	Relu	(None, 16)	528
Dropout		(None, 16)	0
Dense	sigmoid	(None, 1)	17
Total params			397,729
Trainable params			396,897
Non-trainable params			832

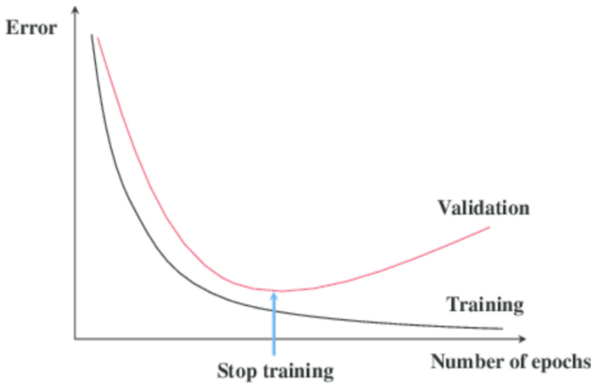


Figure 17. Early stopping rule in term of prediction error

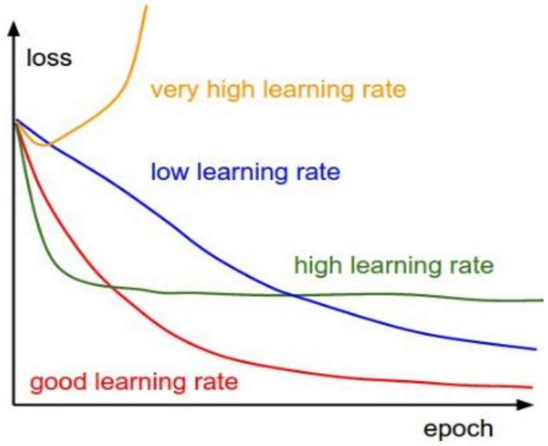


Figure 18. Early stopping rule in term of learning rate

When training and assessing the model, several functions, such as "Early-Stopping," were utilized to reduce the impact of the overfitting issue. Throughout training, it keeps track of the model's performance for each epoch in a constant validation set. The training is stopped before overfitting occurs if the model training is not optimized through iterations. Its general equation is:

$$F_p(x) = F_y\ y\ d\ p(y|x), x\ \in X \qquad (3)$$

where p(y|x) is the conditional distribution at x induced by p. The early stopping rule is shown in Figure 17.

The Keras function "ReduceLROnPlateau" was also utilized to optimize the learning rate during training for the similar reason. One of the most important hyper-parameters to modify is learning rate because it has a significant impact on how effectively the model behaves. As demonstrated in Figure 18, if the optimizer is unable to converge at all loss gradients, a problem may arise. The gradient equation is as follows:

$$g = \frac{1}{m'}\ \nabla\theta\ \sum_{i=1}^{m'} \mathrm{L}(\mathrm{x}^{(\mathrm{i})}, y^{(i)}, \theta) \qquad (4)$$

As a result, the loss function is employed in order to improve the learning algorithms. The training and validation sets are used to calculate the loss function's value, which is then evaluated in light of how well the model did on these two sets. By the value of the loss, we determine how poorly or how well the model performs after each iteration of improvement. The general cross-entropy loss equation is:

$$L(\theta) = -\sum_{i=1}^{k} y_i \log(\widehat{y_i}) \qquad (5)$$

Since accuracy is often dictated by the model's parameters and is calculated as a percentage, accuracy is used to assess how well the learning model is performing. It evaluates how effectively the model's predictions match the actual outcomes.

4.1. Comparison of the proposed CNN-based models

First, the accuracy of the proposed CNN-LSTM model was compared with that of four well-known CNN-based models (VGG-16, ResNet, GoogleNet, and MobileNet). The VGG-16, ResNet, GoogleNet, MobileNet, and proposed CNN-LSTM models were used to build the driver drowsiness model, and they were all trained using the same dataset. Table 3 and Figure 16 illustrate the comparison results. These findings lead us to the conclusion that the proposed CNN-LSTM model has substantially greater prediction training and testing accuracies than that of the famous CNN models.

Table 3. Accuracy comparison of the CNN-based models and proposed CNN-LSTM model

Model	Training Accuracy	Testing Accuracy
VGG-16 [23]	92.46%	62.19%
ResNet [23]	56.09%	49.22%
GoogleNet [15]	66.19%	55.80%
MobileNet [26]	56.54%	41.49%
Proposed (CNN-LSTM) Model	98.3%	97.31%

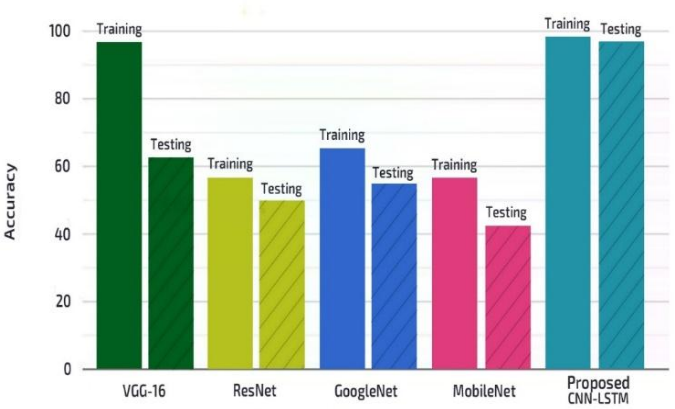


Figure 19. Training and testing accuracy comparison of the CNN-based models and the proposed CNN-LSTM model

The training accuracy are: 92.46%, 56.09%, 66.19%, 56.54%, and 98.3% for VGG-16, ResNet, GoogleNet, MobileNet, and proposed CNN-LSTM, respectively. While the testing accuracy are: 62.19%, 49.22%, 55.80%, 41.49%, and 97.31%. Therefore, we will continue our next comparisons with the recent work using the proposed CNN-LSTM model only.

4.2. Comparison of state-of-the-art models and the proposed CNN-LSTM model

The proposed CNN-LSTM model's accuracy is also compared to the state-of-the-art models that used the same dataset, NTHU. The training accuracy rankings were as follows, from lowest to highest (as reported in their work): [19] achieved an average score of 80%, followed by [39] at 82%, [20] at 83.3%, Dua et al. [9] at 85%, [4] at 85.62%, and D. Venkata et al. (2020) at 88.3%. The accuracy of the proposed CNN-LSTM model was 98.3%. The proposed CNN-LSTM model is compared to the four pre-trained CNN models, namely VGG-16, ResNet, GoogleNet, and MobileNet, in Table 4. Figures 17 and 18 display the accuracy and loss of the model over epochs.

Table 4 shows that, when compared to state-of-the-art approaches, the proposed CNN-LSTM model has achieved the highest degree of accuracy. The best results were achieved by applying the proposed CNN-LSTM model to the dataset and utilizing optimization in validation, augmentation, generator conversion, and Keras functions like (Early-Stop, Checkpoint, ReduceLearningRateOnPlateau, SGD optimizer). In addition to accuracy, we also provide information on loss, precision, recall, and F1 Score for both training and testing. The training yielded the following results: accuracy of 98.3%, loss of 0.0619, precision of 0.9819, recall of 0.9890, and F1 score of 0.9894. Test accuracy was 97.31 percent, validation loss was 0.0859, precision was 0.9682, recall was 0.9835, and F1 score was 0.9758.

In table 4, the CNN-LSTM model performed much better than the next-best model (SVM) [32], which had an average accuracy of 88.97%, by a difference of 8.34 %. Additionally, two significant figures are introduced: Figures 20 and 21 illustrate the model accuracy and loss over epochs, respectively. These data are useful in examining the stability of the model's performance in proportion to the number of epochs. As evidenced by the findings in these figures, our model was able to decrease the training/validation loss in a limited number of epochs, shortening the training period (as is evident in Fig. 21). In a similar manner, the CNN-LSTM quickly increased the training and testing accuracy (see the plots in Fig. 20).

It is crucial to assess the model's performance in the presence of such constraints because the NTHU dataset includes a variety of driving scenarios, including head bending and scarf occlusion. Other scenarios include wearing glasses or not, driving at night or day, and wearing sunglasses. We examined each of these cases' model performance individually. The proposed model's accuracy for each class is presented in Table 5.

Table 4. Accuracy of recent previous models VS the proposed CNN-LSTM model

Model	Accuracy
CNN [19]	80%
CNN and pre-trained on ImageNet [39]	82%
Lightweight CNN [20]	83%
CNN-based models: AlexNet, VGG-FaceNet, FlowImageNet, and ResNet [9]	85%
HOG and NB [4]	85.62%
SVM [32]	88.97%
Proposed (CNN-LSTM) Model	98.3% for training and 97.31% for testing.

Table 5. Accuracy of the proposed CNN-LSTM model under five driver situations

Class Name	Glasses	Night No Glasses	Night Glasses	No Glasses	Sun Glasses
Accuracy	98.23	98.09	97.46 %	98.51	97.92

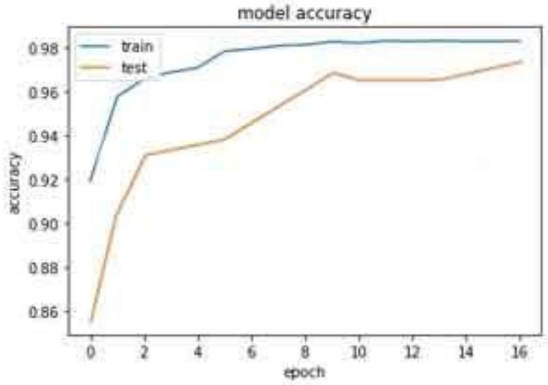


Figure 20. Model accuracy through epochs.

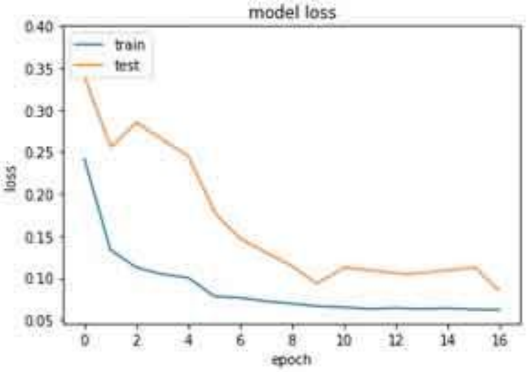


Figure 21. Model loss through epochs.

4.3 Results Discussion

This study concentrated on predicting drowsiness of the drivers through videos captured during his drive. Using a well-known dataset, we conducted tests to verify the proposed prediction models (NTHU video dataset). The following outcomes are impressive when comparing the performance of the proposed model to past studies on the same dataset:

- 98.3% Accuracy, 0.0619 Loss, 0.9819 Precision, 0.9890 Recall, and 0.9894 F1 Score were the results of the training.
- Results showed a 97.31 percent accuracy rate, a 0.0859 testing loss rate, a 0.9682 validation precision rate, a 0.9835 testing recall rate, and a 0.9758 testing F1 score.

It is clear from Table 4 that our proposed CNN-LSTM model outperforms past attempts in the literature on the NTHU video dataset in terms of accuracy. This is as a result of the successful extraction of the dynamic features of the visuals in the video stream, which allowed the CNN model and LSTM model to be coupled to more precisely classify the driving situation. The model's performance during the training and validation epochs, shown in figures 17 and 18, has shown to be steadily convergent to the right findings as seen by the increase in accuracy and decrease in loss as the number of epochs grew.,

Furthermore, since we trained the model under a variety of driver visual circumstances, including both with and without glasses, at night with and without glasses, and with sunglass wear, the accuracy of the model under these conditions was very precise, as can be seen in table 5. The "No Glasses" videos, which represent the ideal driving situation, had the highest accuracy. The classification accuracy was lowest for "With sunglasses," which obscure the driver's eyes and, as a result, degrade the features picked up by the model. However, it is obvious that the eyes are not the sole feature; additional aspects may include the mouth's form, the head's axis, and so forth. It is possible to draw conclusions from an analytical study of the significant aspects that the CNN model and LSTM model automatically translate into dynamic actions.

The proposed CNN-LSTM model significantly improved all of the assessment parameters. As a result, we may draw the conclusion that the overall performance improvement brought about by combining the CNN model with the LSTM is promising and promotes its use in real-time applications.

5. CONCLUSION

Between 25% and 50% of all traffic accidents are attributed to drowsiness, according to numerous recent study studies. According to a research published by the AAA Foundation for Traffic Safety [6], 330,000 vehicle accidents are attributed to intoxication each year. According to the study, around 6,400 of these crashes' 110,000 injuries result in fatalities. Due to this, a number of academics have tracked this issue as a classified problem to determine whether or not the driver is sleepy.

This study uses the NTHU video dataset to propose and assess the CNN-LSTM deep learning model for predicting driver drowsiness. Comparing the final proposed model to earlier research addressing the same subject, it has the highest level of optimum accuracy. The precision attained reached 98.3% for training and 97.31% for testing.

As for the possible future development, we advise implementing the proposed model to the AI embedded system chips, which may then design and implement an entire system directly on vehicles. This would enable the real-time prediction of driver drowsiness while driving, inform the driver of drowsiness, or immediately stop the car if the driver actually experiences drowsiness. This will help to prevent many accidents on the road that are brought on by tired and drowsy drivers. We can also test the performance of the proposed model on different datasets.

Acknowledgments

The authors would like to thank “National Tsing Hua University” for sharing the (NTHU) Driver Drowsiness Dataset which contributed to the success of this research.

The authors would also like to thank Dr. Mohit Dua (2021) for sharing his work in order to contribute to the success of this study.

Funding

This research has not received any type of funding.

Conflicts of Interest

The authors declare that there is no conflict of interest.

REFERENCES

1. Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., & Hariri, B. (2014). YawDD: A yawning detection dataset. In Proceedings of the 5th ACM multimedia systems conference, 24-28.

2. Abbas, Q. (2020). FatigueAlert: A real-time fatigue detection system using hybrid features and Pre-trained mCNN model. International Journal of Computer Science and Network Security, 20.1, 70-78.
3. Aydemir, G., Kurnaz, O., Bekiryazıcı, T., Avcı, A., & Kocakulak, M. (2021). Driver Drowsiness Detection using MobileNets and Long Short-term Memory. IEEE 13th International Conference on Electrical and Electronics Engineering (ELECO), 220-223.
4. Bakheet, S., & Al-Hamadi, A. (2021). A framework for instantaneous driver drowsiness detection based on improved HOG features and Naïve bayesian classification. Brain Sciences, 11.2, 240.
5. Computer Vision Lab, National Tsuing Hua University, Driver drowsiness detection dataset. http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/.
6. AAA foundation for Traffic Safety, accessed Jan 2021, "https://aaafoundation.org".
7. Chui, K. T., Zhao, M., & Gupta, B. B. (2021). Long Short-Term memory networks for driver drowsiness and stress prediction. Intelligent Computing and Optimization, Advances in Intelligent Systems and Computing, Springer, Cham, 1324.
8. de Nauroisa, C. J., Bourdina, C., Stratulatb, A., Diazb, E., & Verchera, J. (2019). Detection and prediction of driver drowsiness using artificial neural network models. Accident Analysis & Prevention, Elsevier, 126, 95-104.
9. Dua, M., Shakshi, Singla, R. Raj, S., & Jangra, A. (2021). Deep CNN models-based ensemble approach to driver drowsiness detection. Neural Computing and Applications, 33.8, 3155-3168.
10. Dwivedi, K., Biswaranjan, K., & Sethi, A. (2014). Drowsy driver detection using representation learning. In Proceedings of the IEEE International Advance Computing Conference (IACC).
11. Faraji, F., Lotfi, F., Khorramdel, J., Najafi, A., & Ghaffari, A. (2021). Drowsiness detection based on driver temporal behavior using a new developed dataset. https://arxiv.org/pdf/2104.00125.pdf
12. Gite, S., Pradhan, B., Alamri, A., & K. Kotecha, (2021). ADMT: Advanced Driver's Movement Tracking System Using Spatio-Temporal Interest Points and Maneuver Anticipation Using Deep Neural Networks. IEEE Access, vol. 9, 99312-9 9326.
13. Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C-K., & Stanley, H. E. (2000). Google team, "Google Colaboratory", ResearchGate, September 2019.
14. Google team, "TensorFlow 2.0 and Keras", ResearchGate, September 2019.
15. Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R., & Shao, X. (2017). Village building identification based on ensemble convolutional neural networks. Sensors, 17.11, 2487.
16. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
17. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the 30th IEEE conference on computer vision and Pattern Recognition.
18. Institute of Medicine. Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem, Washington, DC: The National Academies Press; 2006. https://www.nhtsa.gov/risky-driving/drowsy-driving
19. Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., & Jiang, S. (2018). Real-time driver drowsiness detection for android application using deep neural networks techniques. Procedia computer science, 130, 400-407.
20. Jabbar, R., Shinoy, M., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. (2020). Driver drowsiness detection model using convolutional neural networks techniques for android application. https://arxiv.org/abs/2002.03728
21. Jain, V., & Learned-Miller, E. (2010). FDDB: A benchmark for face detection in unconstrained settings. UMass Amherst technical report, 2.6.
22. Jain, A., Koppula, H. S., Soh, S., Raghavan, B., Singh, A., & Saxena, A. (2016). Brain4Cars: Car that knows before you do via sensory-fusion deep learning architecture. arXiv:1601.00740. [Online]. Available: https:// arxiv.org/abs/1601.00740.
23. Kurama, V. (2020). A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet. https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet.
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In Proceedings of the European conference on computer vision, Springer, Cham, 21-37.
25. MRL Eye Dataset, January 2018. [Online]. http://mrl.cs.vsb.cz/eyedataset (accessed Jan. 20, 2021).
26. Niu, Q., Teng, Y., & Chen, L. (2019). Design of gesture recognition system based on Deep Learning. In Journal of Physics: Conference Series, IOP Publishing, 1168.3, 032082.
27. 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. Iran Journal of Public Health, 47(9): 1371–1378.
28. Reddy, B., Kim, Y., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
29. Saradadevi, M., & Bajaj, P. (2008). Driver fatigue detection using mouth and yawning analysis. International journal of Computer science and network security, 8.6, 183-188.
30. Shakeel, M. F., Bajwa, N. A., Anwaar, A. M., & Sohail, A. (2019). Detecting driver drowsiness in real time through deep learning based object detection. International Work-Conference on Artificial Neural Networks: Advances in Computational Intelligence, Springer, 283–296.
31. Song, F., Tan, X., Liu, X., & Chen, S. (2014). Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. Pattern Recognition, 47.9, 2825-2838.
32. Subbaiah, D. V., PadalaK, P., & Rao, K. V. (2020). A Novel Approach for Detection of Driver Drowsiness Using Behavioural Measures. Lecture Notes in Networks and Systems book series, Springer, LNNS 134, 403-414.
33. Tagad, S. N., Randhave, M. R. & Khan, M. M. (2020). Implementation Paper on Vehicle Drowsiness Detection System. ISSN-2349-5162.
34. Terzano, M. G., Parrino, L., Smerieri, A., Chervin, R., Chokroverty, S., Guilleminault, C., Hirshkowitze, M., Mahowaldf, M., Moldofskyg, H., Rosah, A., Thomas, R., & Walters, A. (2002). Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. Sleep medicine, 3.2, 187-199.
35. The normal sleep cycle. (2000). University of Medicine and Dentistry of New Jersey; New Jersey Medical School.
36. Tibrewal, M., Srivastava, A. & Kayalvizhi, R. (2021). A deep learning approach to detect driver drowsiness. International Journal of Engineering Research & Technology (IJERT), 10.5.
37. Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., & Movellan, J. (2008). Automated drowsiness detection for improved driving safety. https://inc.ucsd.edu/mplab/users/marni/pubs/Vural_icat08.pdf
38. Walling, D. (2020). Sleep and the effects of sleep disruption, sleep foundation, Updated July 28.
39. Wijnands, J., Thompson, H. H., Nice, K., & Stevenson, M. R. (2020). Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks. Neural Computing and Applications, Springer, 32.5, 9731–9743.
40. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology, 9, 611-629.
41. Xiaofeng, L., Xia, J., Cao, L., Zhang, G., & Feng, X. (2021). Driver fatigue detection based on convolutional neural network and face alignment for edge computing device. Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, SAGE, 235.10-11, 2699-2711.