**Driver Drowsiness Detection using Behavioral Measures and Deep Neural Networks Techniques.**

**Chapter 1: Introduction**

Road traffic crashes now represents the eighth leading cause of deaths globally **[1]**. The World Health Organization (WHO) estimated that road accidents claim nearly 1.35 million lives each year and cause up to 50 million injuries worldwide **[1]**. The US National Highway Traffic Safety Administration has estimated approximately 100,000 crashes each year are the direct result of drowsy driver. This resulted in estimated 1,550 deaths, 71,000 injuries and $ 12.5 billion in monetary losses **[2]**. Driver drowsiness is one of the leading causes of road accidents. This is also confirmed by a study conducted by the AAA Foundation for Traffic Safety, which estimated that 7% of all crashes in which a vehicle was towed from a scene, 13% of all crashes that result in hospital admission, and 16-21% of all fatal crashes involve a drowsy driver or lack of sleep. These figures indicate that majority of the road accidents occurred due to drowsy drivers are fatal in nature. Thus, to reduce the number of road accidents and make driving a safer experience, a real-time unobtrusive driver drowsiness detection system is required. Which can detect the impaired state of the driver and generate an alarm if required.

Several factors can be critical to driver’s operational state such as time of the day, alcohol level in blood, fatigue, drowsiness etc. This study focused on one specific impaired state only: drowsiness. Drowsiness can be described as an intermediate state between alertness and sleep. Reduced alertness and tendency to fall sleep are major characteristics to identify drowsiness. Unfortunately, drowsiness cannot be measured or calculated directly, it has to be estimated. Various drowsiness estimations techniques are reviewed in this paper. These methods can be classified into four categories depending on the source of information: subjective, physiological, vehicle-based and behavioral measures. In recent years Karolinska Sleepiness Scale (KSS), ten points graded questionnaire to estimate drowsiness has become the most used subjective measure. It is widely used as an instrument to annotate and validate the labels for drowsiness related datasets through self-assessment. It was also found that none of these feature families is unanimously considered as clear indicator of drowsiness. Each measure has its own limitations and cannot be used as a standalone feature to design real-time systems that is both reliable and unobtrusive. Moreover, there is no direct relationship between these features and driver’s operational state, which is why machine learning techniques were often used with hybrid data in these systems. However, collecting vehicular and physiological data requires simulators, sensors and additional equipment. Considering the scope of this research, behavioral features were deliberately chosen as they can be recorded through unobtrusive recording techniques in the car.

Drowsiness estimation is naturally a difficult task and developing reliable detection systems has been an academic and industrial challenge. Automobile companies such as Mercedes-Benz, Volvo and Nissan implemented vehicular-based features to develop alert systems in their cars. But these systems are often limited to luxury vehicles and not accessible to general drivers. The aim of the current study was to develop and evaluate a model using deep neural networks which can detect driver’s impaired state using behavioral features. The focus was on using drowsiness specific public video datasets only. Throughout literature review it was observed that simple facial features such as blinking rate and eye closure are poor indicators by themselves. Hence, they were often compensated with physiological or vehicle-based measures. Four handcrafted facial features were used for training and testing. These features proved to be more powerful because they combined multiple facial expressions such as eye closure, eye state and yawning in just four elements. In absence of any hybrid data head pose information was used to increase the performance of final proposed model.

Multiple machine learning models starting from simple linear models to advance neural networks were trained on these feature sets. After deriving baseline for results, Recurrent neural networks were chosen to train final model as they accounts sequential data. Finally, this study put forward two hypotheses. First, it hypothesized that it is possible to classify the operational state of the driver as drowsy or alert using facial features such as eye and mouth states. Second, it hypothesized that adding head pose information will improve the overall performance of the model. To recap, the objective of this paper was to develop robust and reliable drowsiness detection system using deep learning and behavioral features. The remainder of this paper is organized as follows: In chapter 2, the literature review of different measures and classification methods is presented. Chapter 3 and 4, consist of proposed methodology and performed experiments. The computational results are discussed in chapter 5 followed by conclusion in chapter 6.

**Chapter 2: Literature review**

**Driver Drowsiness Detection Systems**

Driver drowsiness detection systems (DDDS) continuously analyze multiple features and attributes of the driver and/or vehicle to detect the driver drowsiness. DDDS can be classified based on the measures adopted to determine the level of drowsiness. These measures can be grouped into four categories: (i) Subjective Measures, (ii) Physiological Measures, (iii) Vehicle-based Measures, and (iv) Behavioral Measures. This section provides a review of these four measures, among which first is measured through questionnaire and remaining three by means of various sensors.

**Subjective Measures**

Subjective measures are mainly used as a reference point to evaluate the level of drowsiness. These measures include ratings from the driver or an observer. Driver is asked to verbally describe the state of drowsiness using given scale at fixed intervals. For driver self-ratings, various questionnaires are available such as Karolinska Sleepiness Scale (KSS) **[1]**, the Stanford Sleepiness Scale (SSS) **[2]**, the Epworth Sleepiness Scale (ESS) **[3]**, or the Visual Analog Scale (VAS) **[4]**. For observer ratings, a trained medical professional analyzes the driver drowsiness either in real-time **[5]** or through recorded videos **[6]**. Then compare the state of the driver with given sleep indicators in the facial region.

|  |  |  |
| --- | --- | --- |
| **KSS** | **Verbal Description** | **Vigilance stage** |
|  |  |  |
| 1 | Extremely Alert |  |
| 2 | Very Alert | Alert |
| 3 | Alert |  |
|  |  |  |
| 4 | Fairly Alert |  |
| 5 | Neither Alert nor Sleepy | Partially alert |
| 6 | Some signs of sleepiness |  |
|  |  |  |
| 7 | Sleepy, but no effort to keep alert | Drowsy |
| 8 | Sleepy, some effort to keep alert |  |
| 9 | Very Sleepy, great effort to keep alert | Sleep onset |

**Figure 1.** Karolinska Sleepiness Scale.

The most used scale to estimate drowsiness is Karolinska Sleepiness Scale. KSS is a nine-point scale that uses verbal descriptions at fixed intervals to estimate drowsiness **[1]**. Various studies found that high blink duration, frequent lane departures and other physiological changes are more prevalent in subjects with KSS rating between 5 and 9 **[10]**. Verbal descriptions and corresponding scale ratings used in KSS are listed in Table I. The author of **[7]**used EoG data and corresponding KSS ratings collected every 5 minutes to detect drowsiness. Similarly, Portouli *et al.* evaluated EEG data by estimating drowsiness through KSS and a trained medical professional **[8]**. Various methods that used KSS measures to detect drowsiness are listed in Table II.

Subjective measures give good indicators in simulated environments, but they are not practical in real driving scenarios. Drivers need to answer few questions approximately every 5 minutes, this can alert the driver and reduce the level of drowsiness. It is also dangerous for the driver to provide feedback while driving. Therefore, while subjective measures are useful in determining drowsiness in simulated conditions, other measures may be better suited for real environment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Year** | **Scale and interval** | **Measure used** |
| Hu et al. [7] | 2009 | KSS (5) | Physiological (EOG) |
| Portouli et al. [8] | 2007 | KSS (5) | Physiological (EEG) |
| Sommer et al. [9] | 2010 | KSS (2) | Vehicular (Variation of Lane Position) |
| Ingre et al. [10] | 2006 | KSS (5) | Behavioral (Eye Blink Duration) |

**Table 1.** Drowsiness detection systems based on subjective measures.

**Vehicle-based Measures**

Another method to measure driver drowsiness is to detect abnormalities in the driving pattern. Various parameters such as steering wheel movements, standard deviation of lane position, braking pattern, movement of acceleration pedal and vehicle speed are used to observe the changes in the operational state of the driver **[x]**. These measures are determined using sensors on different vehicle components. The two most used vehicular measures are steering wheel movements and standard deviation of lane position.

Steering Wheel Movements (SWM) are measured using sensor mounted on steering wheel component **[x]**. These movements include standard deviation of steering wheel angle, yaw angle, steering wheel rotation velocity and steering wheel grip. SWMs help in understanding the driver’s steering behavior. Drowsy drivers found to make fewer micro-corrections on the steering wheel compared to the alert drivers **[11]**. However, while changing lanes drivers make extreme steering wheel movements. To avoid these reading researchers considered only small SWMs (0.5° to 5°) which are made to adjust the lateral movement within the lane. Hence, small SWMs are calculated to determine the driver drowsiness and generate an alarm if required. The author of [], use steering wheel angle and yaw angle to detect drowsiness in real driving conditions. First, they investigated both features in drowsy and alert conditions. Then calculated the Approximate entropy (ApEn) features on time series window. These features were then transformed linearly and fed to Back propagation neural network classifier to classify the drivers state. Commercial car companies like Renault and Nissan used SWM methods but found that their reliability is dependent on various factors such as driving expertise, car condition, and geometric road conditions **[12].**

Standard Deviation of Lane Position (SDLP) is another vehicle-based measure used to detect drowsiness. In simulated conditions, a software is used to generate SDLP and in real conditions a front mounted camera is used to track vehicle’s position with respect to center lane of the road. *Ingre et al.* derived numerical statistics based on SDLP and reported that SDLP (meters) increases with increase in KSS ratings **[10]**. However, this relation is not true for all participants. SDLP method is had its own limitations such as lane markings, lightning and weather conditions. In addition, it can also be impacted by driving under the influence of alcohol or depressants. **[13]** Various methods that use vehicle-based measures to detect drowsiness are listed in Table III.

Vehicle-based measures are non-intrusive and are good indicators for detecting drowsiness in simulated conditions. However, their reliability is influenced in real conditions due to various stated factors. Therefore vehicle-based measures are poor indicator to determine drowsiness as a standalone metric. Moreover vehicular-based measures are not specific to drowsiness only.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Author** | | **Year** | | **Measures** | | **Methods and classifiers** | | **Accuracy** |
| *Tansakul et al. [16]* | 2016 | | Blink rate, Driving behavior | | PCA, Haar cascade classifier, PERCLOS, Template matching | | 89% | |
| *P.M. Forsman et al. [17]* | 2013 | | Steering wheel variability, Lateral lane position variability | | PCA, Psychomotor vigilance test (PVT) | | NA | |
| *Li et al. [14]* | 2017 | | Steering wheel angle | | SWA-based fatigue detection method, binary classifier | | 78.01% | |
| *Li et al. [15]* | 2017 | | Steering wheel angle and yaw angles | | Approximate entropy (ApEn) features and backpropagation | | 88.02% | |
| *Ingre et al. [10]* | 2006 | | --------- | | Behavioral (Eye Blink Duration) | | NA | |

**Table 2.** Drowsiness detection systems based on vehicle-based measures.

**Physiological Measures**

Vehicle-based and behavioral measures become apparent only after driver gets drowsy and starts to show the signs of sleepiness, which is often too late to avoid accidents. Whereas physiological changes start to appear in central nervous system in earlier stages of drowsiness. These variations in physiological activity are objective and more direct measures to detect drowsiness. The most investigated physiological measures utilize information based on cardiac activity (electrocardiography (ECG)) **[18,19]**, brain activity (electroencephalography (EEG)) [18,20,22], ocular activity (electrooculography (EOG)) **[18,22]**, and muscle tone (electromyography (EMG)) **[22]**.

EEG appears to be the most used physiological signal to detect drowsiness as it is the most direct indicator of the nervous system activities **[EEG]**. Various frequency bands of the EEG signal and their corresponding activities are listed in Table IV. EEG proved to be the gold standard to detect drowsiness. However, recording the EEG signals is quite intrusive and requires adhesive electrodes to be in contact with the driver’s scalp. Heart rate also varies significantly between alert and drowsy state **[23]**. Therefore, variations in heart rate, which can be determined using ECG signals, can also be used to estimate drowsiness. ECG signals are comparatively more convenient method and use electrodes placed on drivers’ skin. Heart Rate Variability (HRV) is another feature that is used to detect drowsiness. HRV is the measure of beat to beat changes in heart rate and it can be measured using ECG signals. The ratio of low frequencies (0.04 to 0.15 Hz) to high frequencies (0.14 to 0.4 Hz) in ECG decreases progressively as driver tends to feel drowsy **[24,19]**. EoG signals are used to determine driver drowsiness through eye movement tracking. EoG is the measure of an electric field that generates because of the potential difference between the retina and the cornea. This electric field reflects the orientation of the eye. EoG is very accurate measure to detect drowsiness but it requires fine placement of electrodes on the outer corner of the eyes and in the middle of forehead **[18,22]**.

**Table 3.** EEG bands and their associated activities.

|  |  |  |  |
| --- | --- | --- | --- |
| **Band Name** | **Reference** | **Bandwidth** | **Corresponding activity** |
| Delta band | 10 | 0.5 Hz to 4 Hz | Sleep |
| Theta band | 59 | 4 Hz to 8 Hz | Drowsiness |
| Alpha band | 60 | 8 Hz to 13 Hz | Relaxation |
| Beta band | 62 | 13 Hz to 25 Hz | Alertness |

**Table 4.** Drowsiness detection systems based on physiological measures.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Year** | **Sensors** | **Feature extraction** | **Classifiers** | **Accuracy** |
| *R.N.Khushaba et al.[18]* | 2016 | EEG, ECG, EoG | The Fuzzy MI-based Wavelet-packet Algorithm | LDA, KNN, SVM | 95–97% |
| *P.M. Forsman et al. [19]* | 2013 | ECG | Fast Fourier Transform (FFT) | Neural network | 90% |
| *Li et al. [20]* | 2017 | EEG | Fast Fourier Transform (FFT) | Self-organizing Neural Fuzzy Inference Network | 96.7% |
| *Li et al. [22]* | 2017 | EMG, EoG, EEG | Neighborhood search | SVM | 90% |
| *Vicente et al. [21]* | 2006 | HRV | Time-varying mean HR signal estimation | LDA | NA |

Physiological measures are more direct indicator of the central nervous system activities. Hence, their reliability and accuracy are higher compared to other measures. However, these measures are highly intrusive and vary due to various states such as emotions, workload, stress and fatigue. Thus, each physiological measure has its own limits. HR usually drops while driving specifically when the driver is tired **[23]**. EEG has interpretation problems. Peiris et al. (2005) **[24]** showed that two independent practitioners made different assessments to detect drowsiness using EEG for the same participants. Therefore, physiological measures cannot be considered adequate indicators to detect drowsiness. Various methods that use physiological measures to detect drowsiness are listed in Table V.

**Behavioral Measures**

A drowsy person shows several facial movements such as rapid and constant blinking, frequent yawning, and nodding their head **[7.1]**. Behavioral measures based DDDS observed driver’s face using a camera to detect above mentioned facial characteristics utilizing image processing. These methods evaluated mainly three features: eye movements (eye blinking frequency and eye closure activity) via eye-tracking **[29-32]**, facial expressions (yawning, eyebrow rise, jaw drop, and lip stretch) **[33,34]**, and head positions (head scaling/nodding) **[27,28]**. Facial features that were used to detect driver drowsiness include the following.

1. *Eye blink analysis*: Many studies focused on blinking features such as blink rate and blink duration **[30,31]**. These measures used frequency of the eye blinks to determine the drowsiness. The normal blinking frequency is roughly 10 per minute. Fatigue and drowsiness decreases the eye blink frequency i.e. eyelid remain closed for more time. These ocular measurements proved to be good indicator to detect drowsiness. However, blink frequency and amplitude vary from person to person which can impact the quality of the observing framework **[]**.
2. *Eye state analysis*: Some researchers used PERCLOS (which is the percentage of eyelid closure over the pupil over time, reflecting slow eyelid closures, or “droops”, rather than blinks.) to estimate drowsiness **[29,30]**. These systems achieved close to 100% accuracy using PERCLOS. However, this was only possible when subjects did not wear glasses **[30]**. The main limitation of using these features is the lighting conditions. Normal RGB cameras struggle to perform at night. To overcome this limitation **[30]** used Infra-red cameras for active illumination.
3. *Facial geometry*: Fatigue and boredom can cause frequent yawning, which is described as involuntarily opening mouth wide and inhale deeply due to tiredness or boredom. Sudden changes in mouth’s geometry due to yawning were observed to detect drowsiness. Methods analyzing yawning traits to detect drowsiness measures position of lip corners and mouth shape [33,34].
4. *Head pose*: Head position is a reliable indicator to determine the field of view and current focus of the driver. E. Murphy-Chutorian and M. Trivedi [27] presented a system which detected head roll, pitch and yaw in real time to track the head movement. They used three cascaded Adaboost algorithm for face detection. This system worked well in real time under varying illumination conditions. They used Localized Gradient Orientation (LGO) histograms for normalizing frames and three Support Vector Regressors (SVRs) to adjust deviations caused by scale, position, rotation, and lighting.

The main limitation of using behavioral measures to detect drowsiness is lighting conditions. Normal RGB filter cameras do not perform well under poor lightning conditions or during night []. Researchers typically use infrared cameras to overcome this problem. However, infrared cameras suffer during daytime or bright conditions []. Moreover, partial visibility of the face due to camera position or unusual sitting position can further limit behavioral methods []. Wearing sunglasses can further restrict the detection of facial features used for eye analysis. However, behavioral measures are widely used in drowsiness detection systems because they are non-invasive. A dashboard mounted camera is less distracting and comfortable then scalp mounted electrodes. Various methods that use behavioral measures to detect drowsiness are listed in Table VI.

**Comparative study of all measures and why choose behavioral?**

**Table 5.** Drowsiness detection systems based on behavioral measures.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Year** | **Measure** | **Feature extraction** | **Classifier** | **Accuracy** |
| [32]2 E. M. Chutorian et al. | 27 | Head pose (pitch and yaw) | Three cascaded-Adaboost face detectors and Localized Gradient Orientation histograms | SVRs | Na |
| M. Chutorian et al. [6]2 | 28 | Head pose, Lip corner, Eye state | Three cascaded-Adaboost face detectors and Localized Gradient Orientation histograms | SVRs | NA |
| [8]1 | 29 | PERCLOS | Viola-Jones algorithm | SVM | 99% |
| [43]1 | 30 | PERCLOS, eye closure duration, blink frequency, and 3 others | Two Kalman filters for pupil detection | Fuzzy Classifier | Approx. 100% |
| [45]1 | 31 | Blinking frequency, eye closure duration | Cascade classifiers for face detection and diamond search algorithm for face tracking. | Region mark algorithm | 98% |
| [44]1 | 32 | Eye Closure Duration & Freq of eye closure | Eye Closure Duration & Freq of eye closure and Discrete Wavelet Transform | Neural network | 95% |
| G.M. Bhandari et al. [16]2 | 2014 33 | Yawn frequency | YCbCr Color Space, Canny edge | HAAR classifier | 80% |
| M. Saradadevi, P. Bajaj [17]2 | 2008 34 | Mouth and Yawning | Viola-Jones Algorithm, Adaboost, RDF Kernel | SVM | 81% |

**Table 6.** Comparative study of all four measures.

|  |  |  |  |
| --- | --- | --- | --- |
| **Measure** | **Features** | **Advantages** | **Limitations** |
| Subjective | Questionnaires | Subjective | Impractical in real world |
| Vehicle-based | SWMs and SDLP | Non-intrusive | Unreliable, can be affected by multiple factors. |
| Physiological | EEG, ECG, EoG, HRV | Accuracy | Intrusive |
| Behavioral | Blinking frequency, PERCLOS, eye state and yawning | Non-intrusive | Lighting condition. |

**Classification methods used for DDDS**

The general process followed by driver drowsiness systems using behavioral measures is shown in figure I. Video acquisition is the prime step in every real-time behavioral measures-based driver drowsiness system. Video of the driver is captured using a web cam mounted in the car or the cell phone cameras. Recorded video is divided and stored into a series of images/frames. OpenCV framework is widely used for this purpose as it provides great functionality and support for recording and processing videos. Behavioral methods estimate the drowsiness through the use of stored images/frames to observe features such as eye blinking, yawning and head movements. The first step to measure these facial features is to detect the faces in each frame. Paul Viola and Michael Jones proposed Haar-Features based Cascade Classifier object detection framework in 2001 []. This framework was trained to detect multiple types of objects, but it was primarily focused on face detection. It divides entire frame into a set of black and white windows with different weights. These weights are multiplied with pixel intensity to give Haar features. Once Haar features are obtained multiple classifiers are trained using their values. These individual classifiers are then arranged into cascade formation. Each classifier detects one part of the test image and if successful, pass it to the next classifier. Various drowsiness systems used Haar features based cascade classifier for face detection as it detects faces in different scale and works well in real time on CPU []. However, it struggles with non-frontal images and gives a lot of false predictions []. Viola-Jones detector is nearly two-decades old framework and now, it isn’t the only choice for face detection. Navneet et. al. [] demonstrated that a combination of Histogram of Oriented Gradients (HOG) features and a linear SVM classifier could be used to train highly accurate human detector. HOG detector is a lightweight model and works well with both frontal and side images. It is the fastest model among other object detectors including Haar features and wavelets [].

After detecting and locating a face in the image a region of interest is created within it. The process is followed by extracting dense facial features using different methods such as landmark localization, Histogram of oriented gradients (HOG), and Local Binary Patterns (LBP). After obtaining these features, further processing is applied to convert them into advance features such as head angle, yawning frequency, EAR and PERCLOS. These features are then fed to machine learning algorithms such as K-nearest neighbors [], decision tree [], Support Vector Machines (SVM) [], ensemble methods [], and more recently artificial neural networks [] to classify the driver’ s impaired operational state as drowsy or not? Generally, SVM, HMM and CNN are used as classification technique due to their high classification accuracy in drowsiness systems. Machine learning techniques to classify different levels of drowsiness and existing face detection techniques used in drowsiness systems are now discussed.

**Support Vector Machine**

Boser et. al. first introduced SVMs in1992. SVMs are the supervised machine learning techniques used for the variety of classification and regression problems. SVMs try to find the hyperplane that tends to maximize the margin between training examples of different classes. A great deal of research in drowsiness detection systems has attempted to utilize SVMs for multiple purpose in their work. HOG detectors were used with SVMs as the choice of classifier to extract multiple facial features in the image. The author of [] used HAAR like features and Adaboost classifier for both face and eye detection. After eye region is detected, Local Binary Pattern (LBP) was used to track the face over frames which acts as correcting measure. SVM classifier with RBF kernel was trained to perform eye state analysis on the extracted eye features. The system achieved 98.4% accuracy in eye detection and 100% accuracy in face detection tasks. This system reported to work accurate under varying light conditions, background changes and facial orientation. However, it used low frame rate, which could miss multiple facial expressions. Similarly, [] proposed a drowsiness system which used Viola-Jones face cascade classifier to detect face in the images. An SVM classifier was trained to detect fatigue based on eye states such as open, partially closed, closed. The system reported overall 93.5% classification accuracy. A comparison of drowsiness detection systems using SVMs classifiers are presented in Table VII.

**Table 7.** SVM based driver drowsiness detection systems.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Year** | **Measure** | **Face detection method** | **Classifier** | **Accuracy** |
| B. N. Manu [] | 2016 | Eye closure and Yawning |  | Binary SVM with linear kernel | 94.58% |
| G. J.AL-Anizy et al. [] | 2015 | Eye closure |  | HAAR features with SVM | 99.74 |
| L. Pauly and D. Sankar [] | 2015 | Eye state |  | HOG features with SVM | 91.6% |
| A. Punitha et al. [] | 2014 | Eye state (PERCLOS) |  | SVM with Polynomial and RBF kernel | 93.5% |
| M. Sabet et al. [] | 2012 | Eye state |  | SVM | 98.4% |

**Hidden Markov Model**

Hidden Markov Model (HMM) is a statistical model developed in the late 1960’s and early 1970’s by Leonard Baum and colleagues [28]. HMMs are used to predict the hidden state based on the observed state – call it X. HMM assumes that there is another process Y whose behavior depends on X. The aim is to learn about X (hidden state) by observing Y (observed state). The author of [] used HMM to detect facial features such as wrinkles and bulges by observing face intensity. They used Viola-Jones detector to detect face in the images. After detecting a face, image is cropped where the eyes are most likely located. Then it used Gabor Wavelet Decomposition to extract Gabor features which represents changes in the skin texture such as wrinkles and bulges. This study proposed SVMs classifier for detecting drowsiness in single frames and HMMs for the sequence of frames. Another study [] used Pose Extended—Active Shape model for facial alignment and Markov’s models to train on temporal facial features. Two HMMs were used, one to detect head nodding and other to detect blinking of the eyes. This study used state of the art face model which works even under high occlusion. Similarly, [] used Markov’s chain framework for eye tracking based on color and geometric features of the human face. They used two level Llyod-max quantization to eliminate the effect of change in illumination. This method successfully detects the facial region for different head poses but fails to detect face when nostrils are not visible. Table IX gives a brief review of the HMM based drowsiness detection systems with corresponding face detection methods.

**Table 8.** HMM based driver drowsiness detection systems.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Year** | **Measure** | **Face detection method** | **Classifier** | **Accuracy** |
| I. H. Choi et al. [21] | 2016 | Eye blinking and head pose | Pose Extended—Active Shape model | Two HMM for head pose and eye blink detection. | Nil. |
| Zhang et al. [31] | 2015 | Eye state |  | HMM | 95.9% |
| E. Tadesse et al. [38] | 2014 | Eye closure and head pose | Viola-Jones detector | SVMs and HMM. | 97% |
| Y. Sun et al. [39] | 2013 | Eye blinking |  | SVMs and HMM. | 90.99% |
| A. Bagci and R. Ansari [33] | 2004 | Eye state | Color and geometry of the face. | HMM | 99.7% |

**Convolutional Neural network**

Convolutional Neural network (CNN) is a class of deep neural networks and made up of similar artificial neurons. CNNs can take an input image and assign learnable weights and biases to the different objects in the image for object detection purpose. CNNs became popular choice of classifier for computer vision around 2012, when deep convolutional neural networks performed excellent in object recognition tasks []. Table X lists the various CNN based drowsiness detection systems with corresponding face detection methods.

K. Dwivedi et al. [44] proposed a CNN algorithm with two layers for drowsiness detection. They used popular Viola-Jones for detecting a face in the images. These images were then cropped into 48 X 48 square images to fed first layer of the network which consists 20 filters. The output of the multilayer CNN is then passed to a softmax layer for classification task. This system completely depends on visual features and does not consider head pose which makes it prone to fail. However, the author of [] achieved better accuracy by using a 3D deep neural network. For face detection they used two more filters (Kernelized Correlation filter and Kalman filter) which provided robust face tracking. This system works well even if the subject is constantly changing head positions [44].

**Table 9.** CNN based driver drowsiness detection systems.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Year** | **Measure** | **Face detection method** | **Classifier** | **Accuracy** |
| F. Zhang et al. [46] | 2017 | Eye state | AdaBoost, LBF and PERCLOS | CCN | 95.18% |
| B. Reddy et al. [48] | 2017 | Eye state | Eye state and mouth | MTCN and DDDN | 91.6% |
| A. George and A. Routray [47] | 2016 | Eye gaze | Viola-Jones detector | CNN | 98.32% |
| K. Dwivedi et al. [44] | 2014 | Visual features | Viola-Jones detector | CNN with softmax layer | 78% |

**Chapter 3: Methodology**

**Dataset and pre-processing**

Several existing systems used different datasets and measures to detect drowsiness. A major challenge identified through the reviewed literature is that most of the studies used private datasets. They did not share the videos in public domain for open research and evaluation. As a result, it is difficult to evaluate and compare the quality of these datasets. In view of the scope of this research the focus was on using publicly available datasets only. The NTHU driver drowsiness detection dataset (NTHU-DDD), the ULg Multimodality Drowsiness Database (DROZY), and the UTA Real-Life Drowsiness Dataset (UTA-RLDD) are few extensively used drowsiness specific open datasets. All of these datasets were collected in controlled environment which may not always extend to real life situations. For example, the participants of NTHU-DDD were asked to act drowsy while recording videos instead of collecting data when the participants were really drowsy. This fact raised an open question on the usability of these videos to detect the real drowsiness. The use of simulated data did not seem appropriate for detecting drowsiness, as some of the drowsiness indicators are impossible to imitate consciously. Limited number of subjects is another problem with drowsiness specific video datasets. The current study used Real-Life Drowsiness Dataset (RLDD) created by the University of Texas at Arlington for multistage drowsiness detection purpose []. RLDD consists of substantially larger number of participants than DROZY and NTHU datasets. Additionally, all participants were recorded in both conditions. To record not only easily visible but also the subtle signs of drowsiness participants were recorded in conditions of prolonged waking instead of acting drowsy.

**Figure 2:** Sample frames from UTA-RLDD dataset



RLDD dataset provides 180 RGB videos of 60 participants, each recorded in three different states. Participants were from different genders and ethnicities (30 Indo-Aryan and Dravidian, 10 Caucasian, 8 Middle Eastern, 7 East Asian, and 5 non-white Hispanic). All participants were 20 to 59 years old with a mean of 25 and standard deviation of 6 years. The subjects wore glasses in 21 and had significant facial hair in 72 out of 180 videos. Each video was self-recorded by the participant, using a web camera or their cell phone, keeping it at a distance of one arm length. The camera was placed at such an angle that both eyes and mouth were visible. These instructions were used to mimic the real use case in any car. Videos are around ten minutes long and labelled as 0, 5 and 10 for alert, low vigilant and drowsy classes. These labels were provided by the participants themselves, based on their state while recording each video. KSS scale was used to estimate and validate these labels. But still this approach added a subjective element to the process of deciding these labels. The format, orientation of these videos varies a lot because of the fact that they were recorded under different settings. The dataset has total size of 111.3 Gb because of the HD resolution videos. However, the frame rate was always under 30 fps, which is representative of the frame rate expected of usual smartphone cameras. G. Reza et el. also derived a human judgement baseline through an experiment conducted on 20 subjects. The results of human judgement baseline are presented in following figure. The target is to propose a model which performs better than humans in estimating drowsiness level.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Alert | Low vigilant | Drowsy |
| Alert | 0.63 | 0.25 | 0.09 |
| Low vigilant | 0.33 | 0.45 | 0.26 |
| Drowsy | 0.04 | 0.30 | 0.65 |

**Model preparation**

This section provides an overview of the adopted methodology to train a deep learning model on UTA-RLDD dataset to decide if the driver is drowsy or not. The proposed method will sequentially extract frames from the input videos and try to estimate the level of drowsiness based on head movements and facial features of the subject. Four features were calculated to track the eye and mouth states. In addition, roll, pitch and yaw of the head with respect to the camera were calculated for head pose estimation. Together seven features were used to train and test the proposed system. Figure 2. Shows the flow chart of the entire training and testing process.

**A picture containing clock

Description automatically generated**

**Figure 3:** Proposed model architecture

**Extract videos from the UTA-RLDD**

In the first step, all videos were collected from the database. Entire database is divided into 60 folders, one for each subject. Each folder consists of three videos labelled as 0, 5 and 10 corresponding to alert, low vigilant and drowsy state. For the purpose of binary class classification, the interest was only in collecting videos with label 0 and 10 (alert and drowsy state). In this work 120 videos of around 10 minutes each were used for the complete training and testing process., These videos were recorded size and the format

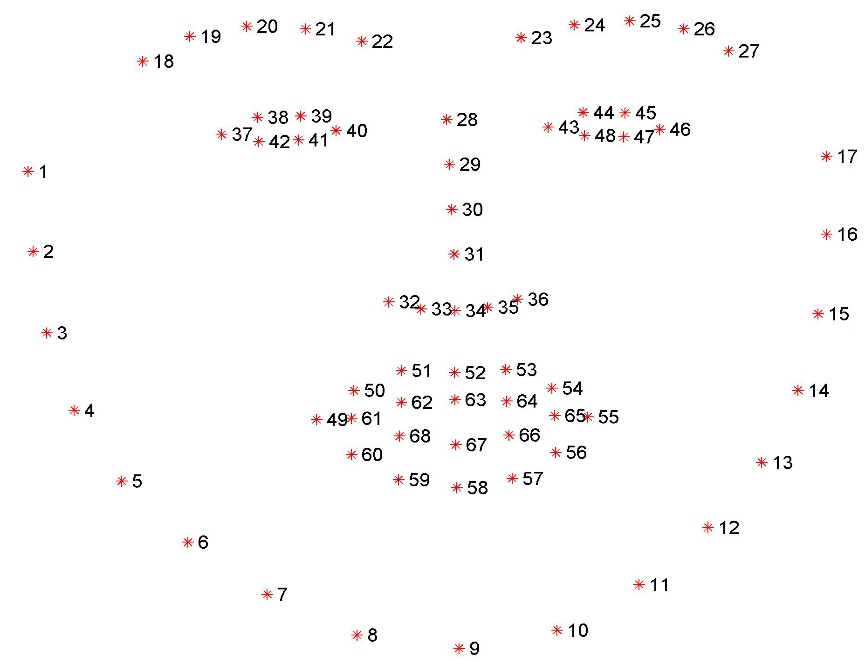
**Extracting images from the videos**

OpenCV library was used to read these videos as it supports all the required formats. OpenCV was again used to convert these videos into images/frames. These images were extracted at a speed of one frame per second. It was noticed that in the beginning of many videos’ subjects try to adjust their sitting position or the camera angle which created occlusion in multiple frames. Similar pattern was noticed in last minutes of the videos as well. To crop the first and last three minutes, CAP\_PROP\_POS\_MSEC parameter was set to 180000 which made videos to start at 3 minutes mark. Then only first 240 frames were collected from each video. These videos are in different orientations which made it difficult to detect a face in extracted frames. FFMPEG was used to identify the orientation through the metadata of these videos. Based on the metadata information videos were rotated if required. Extracted images were then converted into grayscale to reduce the storage size and increase the computation speed.

**Face detection**

In the third step, Dlib library is used to detect a face in the image and then extract the landmark coordinates from it. Dlib is a general-purpose open source SDK written in C++ programming language. It includes a set of independent components for variety of machine learning problems. Its pre-trained models to locate facial landmarks are extensively used in computer vision problems. Initially, OpenCV’s built in HAAR cascade classifiers were used to localize the face in the images. This approach resulted in too many missing or false detected faces. Then Dlib’s pre-trained face detector was used for face detection. It uses Histogram of Oriented Gradients image descriptor and a Linear Support Vector Machine which resulted in a robust face detector. However, both approaches were restricted by the lighting in the frames.

**Figure 2:** Sample frames from UTA-RLDD dataset



In next step, key facial features were detected using Dlib’s inbuilt pre-trained facial landmark detector. There were variety of pre-trained detectors, but all methods essentially target to localize few key facial features only (mouth, eyes, eyebrows, nose and jaw). These models were trained on a training set of labeled images. Initially, images were manually labelled, specifying the pair of X and Y coordinates of region surrounding each facial feature. Then an ensemble of regression trees was trained to estimate the facial landmark position from the pixel intensities only. Unlike HAAR and LBP features based classifiers this method does not require any feature extraction and performs very fast. The current work used the Dlib’s inbuilt 68 facial landmarks shape predictor model. This model was trained on iBUG 300-W dataset, which contains the annotations for 68 facial landmarks. Figure x shows the facial landmark coordinates from the iBUG 300-W dataset. This pre-trained model detects the pairs of x and y coordinates for all 68 facial landmarks. However, for eye and mouth tracking only ten landmarks were used. Similarly, for head pose estimation only six landmarks were used. This combination was able to detect face of 27 subjects in both alert and drowsy states. In total 54 videos of 4 minutes each provided 12960 frames at 1 frame per second.

**Feature Engineering**

In the next step, estimated facial landmark locations were used to localize the important regions of the face and derive dense features. Several features were derived and tested to track the subtle signs of drowsiness. Finally, four facial features and three features for head pose estimation were concluded to be used in the final model.

**Eye aspect ratio (EAR)**

Soukupová and Čech derived a relationship between the length and the width of the eye using equation 1 and termed it as eye aspect ratio. EAR is the ratio of the length of the eye to the width of the eye. Each eye was represented by six pairs of X and Y landmark coordinates. In the equation 1 p1 to p6 are the 6 specified landmark locations of the eye. The numerator of the equation calculates the length of the eye and the denominator calculates the width of the eye. Length is calculated using two sets of points hence, denominator is multiplied by two to get the average. EAR is an estimation of the eye-opening state and was used to determine sudden or consecutive blinking. It remained constant when the eye was opened but fell rapidly to zero when a blink occurred. **First hypothesis was** that when a person gets drowsy, their eyes get smaller and they start to blink faster. Based on this hypothesis, it was expected that if EAR value of a driver falls in successive frames, then model will classify it as drowsy. In order to reduce the computation time only left eye was analyzed.

A close up of a clock

Description automatically generated

**Figure x:** Mouth aspect ratio

A picture containing light

Description automatically generated

**Mouth aspect ratio (MAR)**

Computationally similar to EAR, mouth aspect ratio measures the ratio of the length of the mouth to the width of the mouth. MAR was used to track the geometry of the mouth region without any image processing. MAR stayed near to zero and increased rapidly when the subject opened his/her mouth. The **Second hypothesis was** that when a person gets drowsy, they start to yawn frequently. Based on this hypothesis, it was expected that if MAR of a driver stays higher than usual in successive frames then model will classify it as drowsy.

A picture containing object, clock

Description automatically generated

**Figure x:** Mouth aspect ratio

A picture containing clock

Description automatically generatedA screenshot of a social media post

Description automatically generated

**Pupil circularity (CIR)**

Pupil circularity is just a complementary feature to the eye aspect ratio. However, instead of focusing on the entire eye CIR has its emphasis on pupil only. CIR was calculated using equation 3 where perimeter is the sum of all 6 distances. Similar to EAR, CIR value also remained small for half opened or fully closed eyes due to the squared term in the denominator. A blink will be detected when EAR value falls suddenly but no information will be gained when eye will be only partially opened. In the case of CIR, it was expected that model will predict drowsy if the pupil circularity remains small for a long time.

A screenshot of a cell phone

Description automatically generated

**Mouth aspect ratio over Eye aspect ratio (MOE)**

As the name suggests MOE is simply a ratio of the MAR to the EAR. Both EAR and MAR value were expected to move in the opposite directions when a driver start showing the signs of drowsiness. Proposed model was expected to learn sudden changes in the EAR and MAR ratios to detect drowsy driver. However, these ratios are less prone to subtle changes in the facial geometry and eye state. It was expected that MOE will catch and exaggerate these subtle changes in the EAR and MAR values as both numerator and denominator will move in the opposite directions. Usually MAR showed larger deviation than the EAR values. It was expected that MOE will increase in the case of drowsy driver as it takes MAR as the numerator.

**Euler angles for head pose estimation**

In computer vision language head pose refers to the relative orientation and position of the head with respect to a camera. Head pose is impacted if either the camera or the head changes its position or orientation with respect to each other. When drowsy, a person’s head tends to lean forward due to its own weight. However, then, it becomes difficult to breath and the head tries to regain the normal state, which results in nodding. Nodding and driver’s field of focus have been widely studied to estimate drowsiness. The **Third hypothesis was** that adding head pose information will increase the accuracy of the proposed model. In order to achieve this, three features representing Euler angles (roll, pitch and yaw) of the head were computed.

**Figure 2:** Sample frames from UTA-RLDD dataset

A close up of a person making a face for the camera

Description automatically generated A screenshot of a cell phone

Description automatically generated

2D coordinates of six landmark locations in the image were used to represent head position into 2D coordinate system. These landmarks were nose, the chin, the left corner of the mouth, the right corner of the mouth, the left corner of the left eye, and the right corner of the right eye. These coordinates were then converted into 3D World Coordinates (Model Coordinates in OpenCV docs). The aim was to project six 3D world coordinates into image plane using intrinsic parameters of the camera to measure head’s Euler angles. Motion of every 3D object with respect to a camera can be explained using its translation and rotation vectors. Moving the camera from 3D point A (x, y, z) to a new 3D point B (x’, y’, z’) is called translation. Translation is a three-dimensional vector which represents three degree of freedom (X, Y, Z). Translation vector is equal to (x’ – x, y’ – y, z’ – z). Similarly, rotation represents rotation of the camera in three dimensions. Using Translation and Rotation vectors 3D world coordinates of the head position can be translated into 3D camera coordinates. Then using cameras intrinsic parameters like focal length these 3D camera coordinates were projected into the image plane. However, information about the camera’s focal length was not recorded. The author of [], found a smart hack to estimate focal length of the camera using the width of the image in pixels.

OpenCV’s inbuilt function solvePnP was used to calculate Rotation and Translation vectors using default flags. SOLVEPNP\_ITERATIVE is the default flag which applies DLT solution [] followed by Levenberg-Marquardt optimization [] to get these vectors. In the end, decomposeProjectionMatrix function in OpenCV was used to calculate Euler angles from the rotation and translation vector.

Data analysis

A close up of a device

Description automatically generatedA close up of a device

Description automatically generatedA close up of a device

Description automatically generatedA close up of a device

Description automatically generatedA close up of a device

Description automatically generatedA close up of a device

Description automatically generatedA close up of a device

Description automatically generated

**Chapter 4**

**Experiments**

**Proposed approach**

After collecting and processing the dataset, it was used to train basic classification algorithms. The purpose was to draw a baseline using which stated hypothesizes can be tested. A series of machine learning algorithms were used for modelling, starting from basic classification algorithms such as SVMs, K- nearest neighbors and Logistic regression, and moving on to more complex algorithms containing neural networks. Based on the data size and shape, two classifiers (SVM and Multi-Layer Perceptron) were chosen to draw the baseline for final results. However, these classification models are designed for random predictions and they do not account for sequential data. A function was designed to introduce sequence in the predictions of these classification model. This function average the original predictions with the predictions from the last two frames. Since the dataset was collected in batches for each individual and label remains same for all the frame in a batch, averaging predictions made logical sense. It also helped in eliminating the problem of total random predictions. This approach was good for the baseline models but for the final model a sophisticated algorithm was required which is specifically designed to learn from the sequential order.

**Figure 2:** Sample frames from UTA-RLDD dataset

A screenshot of a cell phone

Description automatically generated

**Introduction to RNNs and LSTMs**

Before a person gets drowsy his/her face starts to show subtle signs of drowsiness. These signs change gradually and are effective in estimating drowsiness. Hence, it was very important that proposed model retain information from previous frames for future predictions. Traditional neural networks do not account for sequential order of the data. They do not retain previous information and inputs are independent of each other. Recurrent Neural Networks (RNNs) solved this problem. They are feedback neural networks in which inputs are related to each other. Their loop like structure enables information to persist in the network for successive predictions. A simple RNN module shown below takes an input xt and outputs ht value. But it also outputs an additional value and pass it through the loop as input to the next module in sequence. Essentially RNN is the copy of same module which unfolds to reveal a chain-like structure where a module is passing some information as input to successive iteration. Standard RNNs are capable of learning short term dependencies very well yet they were not used for detecting drowsiness. Drowsiness is a gradual but slow process, and in the data, recent frames usually had similar values. In this case identical recent frames add no new information and gap between the relevant information is large. Hence, it became very important to use a model which learns from long term dependencies as well.

**Figure 2:** Sample frames from UTA-RLDD dataset

A clock on the wall

Description automatically generated

Hochreiter & Schmidhuber explored the problem of long-term dependencies with RNNs and in 1997 they introduced Long Short-Term Memory networks (LSTMs) a special kind of RNN. LSTMs are capable of learning long-term dependencies and remember information for long period of time. All RNNs are formed like a chain of repeating modules of neural network. In standard RNN this repeating module will be very simple with just one tanh layer. However, in LSTMs instead of one single neural network layer, there are four layers in a module, interacting with each other. Each module has three gates: Forget Gate, Input gate, and Output Gate. LSTM network was used to classify state of the driver because it learns long sequences without gradient vanishing problem faced by traditional RNNs.

**Figure 2:** Sample frames from UTA-RLDD dataset

A picture containing clock, drawing

Description automatically generated A picture containing table, clock

Description automatically generated

**LSTM layers**

The horizontal line running through the top in the diagram below represents cell state. Cell state runs down the entire chain and LSTM layers deliberately removes or adds information to it, regulated by gates.

Forget gate layer: This is the first layer in the module, and it decides which information to let through and which to forget. Forget gate is composed of sigmoid layer and a point wise multiplication operator. Sigmoid layer takes ht-1 and xt as input and generates a value between 0 and 1 for each number in Ct-1 cell state to decide how much of each component should be let through. A value of zero means forget the component while a value of one means let everything pass. For example, in current problem cell state might take gender of the subjects. In that case it will pass through the gender component unchanged until the subject is same but as subject will change forget gate will drop the gender from cell state.

A picture containing clock

Description automatically generated

Input gate layer: This step has two parts. First, a sigmoid layer decides which values in cell state need to update. Next, a tanh layer to create new candidate values Ct, which will be updated to previous cell state. Finally, a pointwise addition of Ct and Ct-1 is done. In continuation with the previous example, this layer will add new gender to the cell state.

A picture containing clock

Description automatically generated

Output gate layer: This is the final step where output is decided. Output will depend on cell state, but it will be filtered through output gate layer. Output gate contains two layers. First is a sigmoid layer which decides what part of cell state to pass through. Then a tanh layer to squeeze the cell state values between -1 to +1 for sigmoid function. In similar example, at this stage network might think that gender is not an indicator of drowsiness and totally discard it.

A picture containing clock

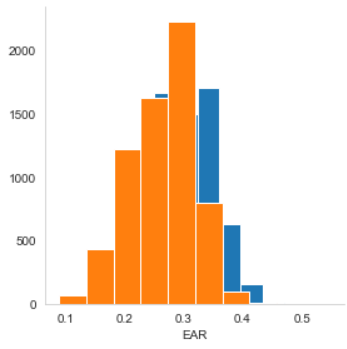
Description automatically generated

**Normalization**

Initially, the entire dataset was randomly split (3:1) into train and test sets. This data was temporal and did not contain any sequence of the frames. The base-line models performed descent, and the prediction accuracy was near to **63%.** However, when the frames were split in sequence by subjects (240 frames per subject per class), models were overfitting. Both SVM and MLP are simple classifiers, hence, structure of the algorithm was not a reason for the overfitting. Through experiments it was observed that models were struggling with new faces. The main reason for this was that size of the eyes varied strongly among the subjects. Due to this, few subjects when alert had smaller EAR and CIR values than other subjects in drowsy state. For example, if person A has smaller eyes than person B and model is trained using B then it will always predict A drowsy. High false-positive rate was another clear indicator for this theory.

Drowsiness detecting is a sequential problem which is impacted by the information in previous frames. In order to use this information, it was required to split the data in sequence only. However, when frames were split sequentially, models tend to overfit. It was thought that scaling features would help to overcome this problem. To normalize the four facial features of each subject, first 3 frames were taken from each individual’s alert video and used as baseline for normalization. First, the mean and standard deviation of each feature for these three frames were calculated. Then these measurements were used to normalize each feature for each subject. When the normal facial features were replaced by scaled/normalized features, the baseline models performed significantly better.

A close up of a logo

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Training**

The input dataset had 12960 records belonging to 27 subjects, it also had 4 or 7 features depending on the experiment. First experiment was performed with baseline models on all seven features after normalization. Entire dataset was grouped by the subjects and then shuffled randomly. Each group had 480 (240\*2) frames which remained in sequence as only subjects were shuffled to add randomness in the data. One 75% of the data was used for training and validation set remaining 25% of the data was used for evaluation. These models were trained using Grid search cross validation to tune the hyperparameters. SVM classifier was trained with ‘rbf’ kernel to deal with nonlinear data, keeping C = 0.1 and gamma = 0.1. Similarly, Multi-Layer Perceptron (MLP) classifier was trained with 120 hidden layers with ‘tanh’ activation unit and SGD solver. In second experiment proposed model was trained on normalized facial features. For LSTM, data needed to be transformed into 3D. LSTM takes data into batches to learn sequential information hence, 12960 frames were divided into 2592 batches of 5 frame each. Similarly label column was also shaped to fit same value for the entire batch. Final shapes of all four datasets sets are shown in following figure.

**Table 2:** Sample frames from UTA-RLDD dataset

A screenshot of a cell phone

Description automatically generated

**Figure 2:** Sample frames from UTA-RLDD dataset

A screenshot of a cell phone

Description automatically generated

It was trained on 50 epochs value with batch size = 5 and Adam optimizer with learning rate = 0.00005. Each batch in training set was sent through a fully connected layer with 1024 hidden units. This layer has sigmoid activation function. Next layer is LSTM layer to learn and retain long and short-term dependencies in the data. This layer has 512 hidden units and an additional dropout layer with 0.5 value for regularization. Next comes three fully connected layers with 256, 32 and 16 units, respectively. RELU activation function is also used to avoid the problem of vanishing gradients. Finally output layer has one unit with tanh function. Figure x shows the data flow and hidden layer design used for proposed model. In the third and last experiment, to test third hypothesis Euler angles for head pose estimation were added to facial features. To execute this training, a laptop was used which had 4 core CPU,16GB RAM, Nvidia GTX 1050ti GPU and Windows 10 operating system. Python was used as primary programming language to use scikit learn API and Keras framework. Keras provides predefined methods for deploying deep neural networks using a low-level training backend. Proposed model was trained using TensorFlow2.0. In the next section, evaluation design and results of these three experiments are discussed.

**Chapter 5:**

**Results analysis and discussion**

These models were trained through supervised learning techniques for binary classification task. Multiple performance metrics suited for classification were used to understand the performance of these models on train, validation and test dataset. The general idea was not to use accuracy as standalone performance metric as it is not a good indicator of the true performance. Hence, these models were evaluated at three different levels. At first level, confusion matrix was used to visualize a clear summary of prediction results. It breaks results into four terms true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The motive was to keep TP and TN high as they represents observations that are correctly predicted. Similarly, efforts were made to drop FN and FP terms which represents observations that are incorrectly classified. Confusion matrix tells not only the errors made by classifier model but more importantly the kinds of errors it made. It overcomes the limitation of using accuracy metric alone. To get a deeper insight into the model area under the receiver operating characteristic curve (AUC-ROC) metric was used at next level. Many classification models also provide probabilistic score for the prediction, if probability is higher than threshold then it is classified as positive else negative. ROC uses these probability scores and multiple threshold values to compute final predictions. High AUC value means model is able to distinguish between the positive and negative class very well. Models were evaluated on both the train and test datasets to generalize results on unseen data and to observe overfitting of the models.

At third and last level, models were evaluated using a combination of the classification accuracy, precision, recall and F1-score. Accuracy tells how many points are classified correctly but it can also be high for a very poor model. Accuracy works well when the cost of FP and FN is similar but in the case of drowsiness detection, cost of FN is much higher. Hence, F1 score was also used which is the weighted average of precision and recall. Precision was used to answer how many of the frames which were predicted drowsy were truly drowsy. Whereas recall was used to answer how many frames were labelled drowsy out of frames which were truly drowsy.

* Accuracy = TP+TN/TP+FP+FN+TN
* Precision = TP/TP+FP
* Recall = TP/TP+FN
* F1 Score = 2\*(Precision \* Recall) / (Precision + Recall)

**Baseline models**

When trained on normalized facial features and head pose features all together, SVM gave 73.7% accuracy. Recall value (0.94) was much higher than precision value (0.66) meaning SVM classifier was classifying maximum frames as drowsy. High false positives (786) also proved this observation. With the help of confusion matrices and AUC values on both the train and test set it was found that SVM was clearly overfitting. Performance on train data was exceptionally perfect but it degraded tremendously on test set. Similarly, MLP was trained on all seven features and it gave significantly better results. This model had maximum accuracy (80.4%) out of all trained classifiers. Both the precision and recall values were very similar, suggesting that model learned from the training data. However, performance on the test data was much lesser than on train data. AUC value on train set was 0.98 and on test set it fell to 0.81. Hence, for MLP classifier errors could not be generalized on unseen data. This performance boost gained over SVM by using simple neural network hinted to use more sophisticated neural networks like LSTM.

A screenshot of a cell phone

Description automatically generated A close up of a map

Description automatically generated

**Figure 2:** Confusion matrix and ROC curve for MLP classifier on test data.

A screenshot of a cell phone

Description automatically generated A close up of a map

Description automatically generated

**Figure 2:** Confusion matrix and ROC curve for SVM classifier on test data.

**LSTM classifier**

Initially, LSTM classifier was trained with only four normalized facial features to test the first hypothesis. Observed precision value of this model was 0.80 which means it successfully predicted most drowsy frames as drowsy. But low recall value (0.327) signified that it had too many false negatives. Same was concluded from the confusion matrix. This was a big point of concern as both the classes were perfectly balanced. As a result, overall accuracy of the model was only 62.5%. But unlike baseline models performance of LSTM classifier remained same on both the train and test set. One major reason for the poor performance was the limited data. Neural networks require huge amount of data to train. In the next step head pose information was merged with existing data which filled feature space with additional information. This resulted in an increase of 14% in accuracy and tremendous improvement in recall value. All metrics were nearly similar for both the train and test set. Hence, results could be generalized on unseen data. High AUC value (0.86) was observed which signified that proposed model learned the data and was able to clearly distinguish between both the classes. Adding head pose definitely improved the model and proved second hypothesis.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Models** |  | **Train** | **Data** |  |  | **Test** | **Data** |  |
|  | TP | TN | FP | FN | TP | TN | FP | FN |
| SVM | 4800 | 4800 | 0 | 0 | 1583 | 894 | 786 | 97 |
| MLP | 4601 | 4675 | 125 | 199 | 1247 | 1455 | 225 | 433 |
| LSTM (4 features) | 404 | 848 | 112 | 556 | 110 | 310 | 26 | 226 |
| LSTM (all features) | 571 | 935 | 25 | 389 | 260 | 251 | 85 | 71 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC on train data** | **AUC on test data** |
| SVM | 0.737 | 0.668 | 0.942 | 0.781 | 1.0 | 0.77 |
| MLP | 0.804 | 0.847 | 0.742 | 0.791 | 0.98 | 0.81 |
| LSTM (4 features) | 0.625 | 0.808 | 0.327 | 0.466 | 0.76 | 0.76 |
| LSTM (all features) | 0.760 | 0.753 | 0.773 | 0.763 | 0.87 | 0.86 |

Cannot trust on simulated data, need real conditions

Hybrid measures and wearables

**Chapter 6**

**Conclusion and future scope**

In this study, various measures and classification methods used to detect drowsiness were reviewed. For drowsiness estimation problem behavioral measures were found to be the most cost efficient and reliable features. Four handcrafted facial features along with three Euler angles and binary labels were used to train multiple classifiers. A Recurrent neural networks-based model which learns and retains long and short-term dependencies in the data was proposed to detect driver’s drowsiness. This model was trained on 1920 batches of 5 frames each. With limited data proposed model was able to achieve 0.86 AUC value and 81

% classification accuracy on test data. This paper illustrated that using RNNs on sequential data for drowsiness estimation results in very robust and reliable classifiers. Error rates in the proposed model are very similar in all three data sets which generalizes the performance of the model on unseen data. Results of LSTM with all features demonstrated that proposed model outperforms human judgement baseline in binary classification task. These results also proved both the hypotheses. The proposed model also has low computational time and storage requirement as it does not perform any image processing to observe key facial features. These characteristics make it suitable for applications on mobile platforms. Yet this work is not fully ready for real-world applications as drowsiness detection systems demand high accuracy and least possible false negatives.

Future research should consider combining physiological measures, such as ECG, with vehicle-based and behavioral measures to form a hybrid feature matrix. Wearable devices can be used to extract ECG in non-intrusive way. Similarly, vehicular data can be generated using driving simulators by closely monitoring driving environment for optimal results. Most studies used classification techniques for detecting drowsiness as it appears. Proposed methodology can also be transformed to convert it into a regression problem. By doing this focus can be shifted towards forecasting drowsiness before it happens.