FYP2

by Joshuaa C

Submission date: 23-Feb-2021 03:15PM (UTC+0800)

Submission ID: 1516000135

File name: TurnItInCopyFYP2.pdf (356.97K)

Word count: 8845

Character count: 44626

Chapter 1: Introduction

1.1 Problem Statement

Driver drowsiness or driver fatigue, is a condition where the driver begins to lose attention on driving due to a lack of rest or from an array of untreated sleep disorders, such as obstructive sleep apnea, insomnia, and narcolepsy (Drowsy Driving, 2020). This condition can not only endanger the driver, but other drivers may become collateral to the situation. This also means that a road user could be a victim to driver drowsiness without suffering its effects for themselves.

Accidents caused by driver drowsiness are not uncommon. In an online article from (Richards, 2020), an accident was caused due to the driver drifting off to sleep on his way to work in the state of Utah, USA. According to the highway patrolwoman, a statement was taken from the driver post-accident, stating that he woke up late for work, causing him to feel drowsy on his journey to his workplace. This incident was reported at around 7 a.m. Another incident was reported by The Star regarding an accident that happened in Bangkok, Thailand (Thailand: Broken railway signal, lack of gate blamed for deadly accident in Chachoengsao, 2020). This incident represents a more tragic effect of driver drowsiness, as a father was killed and his family injured due to a 22-wheeled lorry driver driving recklessly under the influence of drowsiness.

Driver drowsiness does not stem from any bad or illegal influence such as drug and alcohol abuse. It is a condition that can happen at any time a driver is at the wheel.

According to an E-Survey conducted by the ESRA (Goldenbeld & Nikolaou, 2019), in most countries one fifth to one quarter of car drivers report to have driven while having trouble keeping eyes open in the past 30 days within the writing of the given report. Driver drowsiness can be countered by the driver by attempting to take a short rest as soon as possible to avoid being a hazard on the road. However, even this could be a tricky task as some drivers are unable to notice when they are drowsy in time. Even for those who are able to detect their state of drowsiness, they are pressured by other factors such as strict time constraint (i.e working hours, meetings, etc.) or the lack of a suitable rest area.

Drowsiness is a natural condition for humans. In fact, it in itself represents a stage of the human sleep cycle. Sleep can be categorized into three stages; namely active, non-rapid eye movement (NREM) sleep, and rapid eye movement (REM) sleep (Brodbeck, et al., 2012). The second stage of sleep is further divided into three categories, the first two (N1 and N2) representing light sleep and the third (N3) representing deep sleep. In this case, drowsiness is represented by the stage that transitions from the active state to REM, which is N1. It is also the state researchers are commonly focusing on to analyze driver drowsiness.

With this issue in mind, researchers have been experimenting with ways to detect driver drowsiness. These methods are implemented by deriving readings from a range of inputs that could be obtained regarding the driver by the hardware placed within the automobile. Generally, the metrics of current implementations can be summarized into three categories:

1) Vehicular metrics – Inputs that are obtained by the driver that determine the state and movement of the automobile such as steering wheel movements, pedal pressing, signal usage, etc.
2) Physiological metrics – Inputs that are obtained from physiological data from devices such as electrocardiogram (ECG), electrooculogram (EoG) and electroencephalogram (EEG).
3) Facial Behavioral metrics – Inputs that are visually obtained from the features of the driver's face such as blinking, yawning, eye visibility, etc.

The implementations above are coupled with alert systems to notify the driver in the event drowsiness is detected. Standard alert systems include audible chimes from within the vehicle cabin, or visual cues through the dashboard of the vehicle, such as flashing indicators.

Driver sleepiness is a common cause for road accidents globally. It can happen to the best of drivers at the most unexpected moments of a journey. It can however be tackled by notifying the driver in time regarding their drowsy state, in hopes of them deciding to take the necessary precautions. Therefore, by developing a feature for a monitoring system that can detect when the driver is sleepy, it could be used as a potential security feature for many modern automobiles to reduce the rate of road accidents cause by driver drowsiness.

1.2 Research Objectives

The objectives of this FYP are:

- to perform a background study on the current implementations of driver drowsiness detection to study in the detail the developments, paths and problems faced by researchers in furthering this line of safety features.
- to study and classify the facial features required to be learned by machine learning for driver drowsiness detection

1.3 Project Scope

This study is centered around the detection of the drowsy state of the driver using visual inputs such as image and video data.

Chapter 2: Literature Review

As discussed before, current implementations of driver drowsiness detection systems can be generalized into three categories. Each of these categories observe inputs from different focus areas, such as the driver or the vehicle. The input data is then analyzed for patterns which can be used to detect driver drowsiness, which will trigger a response to alert the driver on their current state. The following subtopics will be discussing the three general categories of driver drowsiness detection.

2.1 Vehicular metrics

Studies regarding this form of metric can be found dating back as far as 1994 (Knipling & Wierwille, 1994). It represents some of the earliest interests and implementations of driver drowsiness detection. Detection in this form of metric commonly includes sensors within the vehicle itself to monitor the driver performance to detect the state of the driver and to execute counter-measures if possible.

Current studies and implementations focus on a range of vehicle-based signals that include, but are not limited to, steering wheel activity, monitoring for lane drifting, and pedal states (Yang, Xi, & Wang, 2019). Steering-wheel movement (SWM) is a common indicator in most vehicle-based driver drowsiness detection systems and studies. Due to the understanding that fatigued drivers would make lesser micro corrections and more erratic and obvious macro corrections (Borghini, Astolfi, Vecchiato, Mattia, & Babilonii, 2012), SWM can be applied to detect these observations. The data obtained can be used to detect the current state of the driver at the wheel. A detector that monitors lane drifting has been implemented by observing standard deviations of the position of the cars in their respective lanes (Anund, Kecklund, Vadeby, Hjalmdahl, & Akerstedt, 2008).

Features such as these are then collected and weighted accordingly, before being applied with machine learning techniques, specifically classifiers. Common classifiers used in studies include Support Vector Machines (SVM) and Back Propagation Neural Network (BPNN) (Yang, Xi, & Wang, 2019).

The engineering company Bosch has implemented a driver drowsiness detection system (Driver Drowsiness Detection, 2020) using vehicular metrics, which, in the case of their algorithm, applies data from steering wheel movements. Their algorithm begins recording the drivers steering wheel activity from the beginning of each trip, monitoring for patterns that display signs of change in movements of the steering wheel. The usual signs the algorithm is able to pick up are when the driver is barely steering, or when the steering is influenced by quick and abrupt movements to keep the car study. Other parameters that influence the detection include the length of the trip, turn signal usage, and the time of the day. Through these signs, the algorithm is able to pick up on the driver's wavering concentration on the road. When the frequency of these signs reaches a certain threshold, the algorithm warns the driver by flashing a coffee cup icon on the instrument panel, alerting the driver on their current state.

The perks of implementing a system with these metrics are they are not influenced by the physical cosmetics of the driver (Haupt, Honzik, Raso, & Hyncica, 2011). Wearing masks, dark-shaded glasses or winter clothing will not influence the performance of the monitoring system in any way. The monitoring system would also be non-intrusive, meaning that the data obtained by the system can, and will, only be used for detecting the state of the driver.

As robust as the metric may appear to be, vehicular metrics has its share of disadvantages. Certain situations other than the driver being drowsy may trigger the detector out of mistake, causing high occurrences of false positives or false negatives. For example, micro-corrections or macro-corrections could occur due to the driver driving on unfamiliar roads (Vural, 2009), or even random road events such as potholes and debris on the road. Lane monitoring implementations may also not be optimal as deviation from lanes and erratic steering wheel movements become much more common in dense traffic scenarios, potentially triggering the detector despite the lack of fatigue (Sikander & Anwar, 2019).

2.2 Physiological Metrics

Driver drowsiness can also be detected through a more biological approach, that is by studying the physiology of the driver itself. This form of metric is considered to be accurate (Awais, Badruuddin, & Drieberg, 2017) as it represents biological signals from the driver itself. By observing the driver through the use of this metric, driver drowsiness can be detected.

Currently, the most reliable method in this metric is Electroencephalography (EEG) (Awais, Badruuddin, & Drieberg, 2017), a test that measures one’s brain activity by its electrical signals. The raw signals are obtained by EEG electrodes attached to the scalp, which is then studied by dividing them into different frequency bands, which is then studied to observe the current activity of the brain. Electrocardiogram (ECG) is another test that is commonly to monitor electrical signals from the specimen's heart. The signals, which are displayed as waves, can be observed to detect fatigue.

Signals from various other parts of the body can also contribute in detecting driver drowsiness. According to (Doudou, Bouabdallah, & Cherfaoui, 2018), aside from the brain, different organs in the human body that show obvious signs of alertness or drowsiness of the driver can be observed for drowsiness detection. These modes of measurements include, but are not limited to, electrooculography (EOG); which observes the activity of the eyes, electromyography; which observes the activity and condition of the muscles and nerves of the body, and electrocardiography and blood pressure signals; which observe the current state of a person's cardiac activity. Other physiological signals such as respiration, gastro-intestinal data, electro-dermal activity and even core temperatures of the driver can be used to provide better insights on the current state of the driver and thus, detect any forms of fatigue.

A major advantage physiological metrics may have compared to the other metrics being reviewed in this chapter, is its accuracy. As stated before, this form of metric is observing biological signals from the driver itself. Instead of observing the external physical effects of drowsiness through input receive from the vehicle, the data observed through this metric is received directly from the source, which happens to be the driver.

In addition to that, the data used through this form of metric are universal (Haupt, Honzik, Raso, & Hyncica, 2011). The findings and assumptions that the detection is based on is widely accepted and can be directly applied in other sectors.

However, despite the effectiveness of this form of metric, to be using these methods to detect driver drowsiness in real-world driving conditions may not be feasible. This is due to the expensive and complex equipment needed to conduct these tests (Sheng Yang, Wen Zhong, & Yan Yang, 2017), even more so for long hours of long trips, where driver drowsiness is a common occurrence. This form of metric is also sensitive to external variables such as driver conditions (Haupt, Honzik, Raso, & Hyncica, 2011). The clothes that are worn by the test subjects as well as the external climate, may alter the results from the various test machines, providing inaccurate readings and further inhibiting the rate of successful detection. Health conditions of the driver may also affect the detection if not taken into consideration.

2.3 Behavioral Metrics

Most people who are fatigued while driving show noticeable visual behaviors, particularly observable from their facial reactions. A typical form of response from a person who is drowsy while driving may include swaying of the head, slow eyelid movement and a longer blink duration. An implementation with this form of metric studies these facial behaviors of the driver to determine the level of drowsiness.

Most available studies have been conducted by studying the eye of the driver for features such as rapid blinking, or slow closure of the eyelids (drooping). PERCLOS is the percentage of eyelid closure over the pupil over time and reflects drooping, rather than blinks (Dinges & Grace, 1998). This form of the metric was deemed reliable in many studies to accurately predict drowsiness. Multiple other facial behaviors have been used by researchers, some of them include eyebrow rises, lip stretching and jaw drops.

One implementation of this work studied the tracking of the head and eye of the driver using two cameras (Matsumoto & Zelinsky, 2000). In this implementation Matsumoto and Zelinsky used an image processing board, specifically an IP5000, which houses a wide array of fast image processing functions. Connected to an external camera and a video output monitor, it represented the hardware that they used. A field multiplexing technique was then used on the video feed to create a multiplexed video stream from the analogue phase. With the stereo video feed, an initialization phase is conducted, where the algorithm searches the video feed for a face using a 2-dimensional template. Once detected, the algorithm begins its head pose tracking function where a 3-dimensional template is applied to determine head pose. If successful in tracking the head pose, the algorithm calculates the 3D gaze vector, and returns to the face tracking phase and repeats the process. If the system fails to detect a phase in the face tracking phase, the system reverts back to the initialization phase where it once again attempts to detect a face.

Template matching is a method which attempts to match parts of the live data being received to available templates in the system. These templates are available images taken and readily classified to their respective states (i.e. open eyes, closed eyes, etc.)

(Ullah, Aslam, Ullah, & Martinez-Enriquez, 2018). The templates are then constantly filtered over the input data to find matches in states with the templates and uses these as key points. From this, details such as eye closure times can be calculated and compared to a predefined threshold of values, for which then periods of drowsiness can be detected. In (Krolak & Strumillo, 2012), a similar approach was applied in an implementation to detect eye blink using computer vision. They proposed the idea with intentions of applications with fatigue detection, human – computer interactions and even lie detections. The system was built on a consumer-grade PC connected to a medium quality webcam. The video feed was recorded with small pixel dimensions (320x240) at 28 frames-per-second (fps). The data was then feed through an algorithm which consisted of four main stages. In the first stage, faces were detected from the video feed using Haar-like features and a collection of boosted tree classifiers. From this, the next stage included extracted the region of interest (ROI) from the facial detection, which in this case, where the eyes on the face. This is done based on specific geometrical dependencies which are known about the human face. With the ROI extracted, template matching is then conducted with existing labeled templates to classify the ROI according to the eye-blink classifications.

Another implementation of this work studies and tracks the pupils of the driver (Bergasa, Nuevo, Sotelo, Barea, & Lopez, 2006). The visual data is acquired through a charged-coupled device (CCD). The data is then processed and segmented with respect to the pupils of the driver, and actively tracked in real time. The system also accounts for poor lighting by using near-infrared illumination to illuminate the driver's face for better detection, a feature system which is also discussed in (Krolak & Strumillo, 2012). The benefits this method brings are it does not affect the performance of the driver as well as minimizing the changes in ambient lighting. This also contributes to creating the bright pupil effect, making it easier for an algorithm to study the pupils of the driver for more efficient predictions. Parameters are then derived from the data to detect observable signs of fatigue as stated before such as eye "drooping", smaller eye openings, face pose, frequency of blinking and nodding. The parameters are then summarized together to determine the state of the driver. When a certain threshold is achieved, an alarm will be triggered.

Current implementations of this type of metric in the automobile industry include Seeing Machines (Seeing Machines, 2020), an Australian-based company which uses computer vision for its Driver Monitoring Systems (DMS) to improve road safety. According to their website, their systems have successfully detected 7,507,874 distraction events as well as 158,712 fatigue interventions in the pass year.

A perk of applying this form of metric is that it is non-intrusive as well as most natural as compared to the other metrics as it studies the movement of selected facial features of the driver (Yong Du, 2008). As discussed before, drowsiness is reflected by the behavior of one's face and eyes (Garcia, Bronte , Bergasa, Almazan, & Yebes, 2012). Studying these features adds to the natural aspect of this metric as the data that is derived originates from the instincts of the driver. Despite having a camera monitoring your face over extensive period of time, it does not extract the personal details of the driver, other than the features that can be observed on their face. It also does not require bulky testing machines, other than a dashboard camera and a computing system, which can be neatly stored in the driver's cabin.

Despite the convenience the metric brings, it has a disadvantage when it comes to lighting. To save costs on equipment, normal and relatively cheaper cameras are used as the driver dashboard cameras for the implementations. These types of cameras do not particularly excel at recording in low-lit areas, which causes the recorded data to be noisy and unclear, reducing the ability of an algorithm to observe the facial behaviors of the driver.

2.4 Summary

Driver drowsiness detection can come in any shape and form, none being anymore superior than the other. Each of the aforementioned metrics exist with their own set of advantages and disadvantages. The choice of metric depends on the needs and requirements of the study itself. Table 1 shows a summary of the characteristics of each of the metrics that have already been discussed thus far;

Studies based on facial detection have been extensive throughout the computer vision environment. From security measures to information extraction, facial detection implementations have birthed a wide array of ideas that can potentially change the current norm of convenience and efficiency. Applying this school of thought to driver safety features may not only introduce new paths to accident prevention, but also provide us a greater understanding of our behaviors as human beings.

A vision-based drowsiness detector will be the focus of this project, which is a representation of the facial behavioral metric. As can be summarized from Table 1, the benefits of this metric are that it is non-intrusive and is the most natural form of metric among the stated categories. This metric has also seen multiple approaches to providing results, partially due to the fact that this metric can be applied with minimal hardware involved.

After a comprehensive study on the methods of implementation of this metric, it is understood that a functional drowsiness detector can be implemented using neural networks, specifically Convolutional Neural Networks (CNN). CNN is a popular and efficient manner of applying deep learning methods for data that comes in visual form, such as images and videos. By using CNN as an image classifier, a model could be trained to potentially distinguish between states involved in drowsiness, allowing for further build-up in the implementation to detect drowsiness as a whole.

Chapter 3: Research Methodology

An efficient research progress would require a set of methodologies to follow in order to make good progress as well as to be on track. Below is the planned methodology for this research, which is modelled after the Cross-Industry Standard Process for Data Mining (CRISP-DM) (Chapman, et al., 2000). This methodology was created late in the year 1996 when the data mining market had not seen much attention yet. It is hierarchical process model, which is described by a set of tasks, which have been altered to fit the processes involved in this study.

The following sections describes the phases of the research methodology for this study, which has been adapted from CRISP – DM.

3.1 Literature Review

In the first part of the research, a background study on the field itself will be conducted as well as looking into existing iterations of the idea of driver drowsiness detection. This is step is important to gain the important background knowledge of the topic to conduct the research in an efficient manner, without ignoring sections vital to its developments. Studies on drowsiness detection have been in discussion for many years and many researchers have made important findings regarding state-of-the-art methods. It is imperative to study and understand these developments as to not yield inaccurate conclusions or misuse techniques. With the current knowledge of the state of the area of interest, it will be easier to conduct a study with meaningful results.

3.2 Data Gathering and Understanding

In this phase of the project, data regarding the topic at hand will be gathered to be used in the experimentation in the later phase. The data will be explored and evaluated for features that could be used to detect drowsiness such as eye 'drooping' and rapid blinking. Due to the type of features required for this research, visual data would be required, specifically video data. The data would have to be scrutinized as to understand how it was collected and how it should be dealt with. This is so that the data is not misused, which could potentially lead to adverse results. This step can also be supported by other papers by studying the methods in which they have dealt with their respective data.

3.3 Data Preparation

In this phase, the selected data will be prepared to be fed to our detection model(s). Since the data that will be focused on is video data, proper techniques have to be used in order to convert the video data into image data, as training computer vision models cannot be applied with video data directly. Hence, the video data has to be converted into image data without taking up too much storage space as well as not losing too much data. Once cleaned, the data has to be sorted to subsets such as training and testing sets, as well as validation sets if required.

3.4 Experiment Planning

This part of the research covers on researching existing computer vision models used to detect driver drowsiness. With these models, several tests will be created to develop a deeper understanding on how these models work as well as studying their performances in various scenarios. The experiments will be planned to test various aspects of the model and the results will be recorded for evaluation.

3.5 Experiment Evaluation

The research will then be concluded by conducting the experiments planned on the existing models. The results will then be collected and studied, which will provide us with a deeper insight on the performance of techniques involved. Evaluation metrics such as confusion matrices, recall and precision scores will be calculated and compared to provide support to conclusions to be made from the experiments. Graphs and charts will also be drawn to provide a greater visual understanding of the performance of the techniques in the various tests.

Chapter 4: Theoretical Framework

In this chapter, the frameworks relevant to the project will be discussed in order to have a better understanding of its application in FYP II.

4.1 Convolutional Neural Networks (CNN)

In order to under the functionality of CNN, we need to understand what a neural network is. Neural networks are a paradigm of the human brain, where its conception was based on the understanding of how neurons in the human brain function with input received to give an output. Together with deep learning, neural networks are the current efficient means of providing solutions and insights from unstructured data such as images, video, audio and text blocks. Image recognition and natural language processing are current examples of applications of not just NNs, but its many variations, such as CNNs and Long Short-Term Memory (LSTM).

The architecture of neural networks is based on layers. There are three general types of layers for a neural network, namely the input layer, hidden layers, and output layer.

From the figure above we can see the division of the respective layers. On the far left, the blue nodes represent the input layer. The number of nodes in this layer can vary, depending on the shape of your data (i.e. tuple, 2-D array, image array, etc.). In the middle, the black nodes represent the hidden layers of the neural network. The number of hidden layers varies in each model, depending on the complexity and purpose of the model. On the far right, the green nodes represent the output nodes. The output nodes can also vary depending on the purpose and intended output for the model. The example above represents a fully-connected neural network as each node in a layer is connected to every node in the following layers.

When training a neural network, the data, which is represented commonly by floating points in the network, is passed through from the input layer through the hidden layers, and finally reaches the output layer for prediction. This process is repeated multiple times over the same dataset to train the model. Each pass of the dataset through the model is called an epoch. The way the model learns is by applying weights to its connections. The weights are first initialized by an arbitrary value. These weights are then updated calculated by other metrics within the neural network such as the gradient of the loss and the learning rate. The gradient of the loss is simply the rate of error of a neural network. The learning rate is a small factor that represents the size of a 'step' for a machine to take to reach its goal, which in this case, are optimal predictions.

Each layer in the neural network has an activation function. The purpose of these functions is to take the data from the previous layer, conduct a transformation on it, and pass it on to the next layer. These functions often times return a value within a certain upper limit and a lower limit. There are many forms of activation functions, each with its own functions. Examples of these activation functions are sigmoid activation and rectified linear unit (ReLU) activation. A sigmoid activation function returns a number between 0 and 1, depending on the negativity or the positivity of the value. A ReLU function returns a number between 0 or the input value itself.

Convolutional Neural Networks (CNN), as discussed earlier, is a form of neural networks that has efficiency in dealing with visual input. Image classifications is the common use of this type of model due to its architecture, more specifically, its convolution layer. The input layer for CNN takes in visual data. However, instead of processing the image directly, the model takes in the image as an array of pixels. for a neural network, namely the input layer, hidden layers, and output layers.

The convolutional layer is the main component of a CNN. In the context of CNN, convolution is the act of 'sliding' a filter over the whole image data to expose to the model interesting patterns such as edges or corners, to even full structures such as a leg or a car tire. The transition of detecting objects begins at a basic level such as detecting pixel patterns, and slowly builds up in complexity with each layer in a CNN such as detecting shapes, until a point where real-world objects can be detected.

A filter of a convolutional layer is a matrix that will be passed over the image matrix to manipulate its values. To explain this process, we will be visualizing the convolution of a 5x5 image with a 3x3 convolutional filter.

The filter is passed onto the image (blue) at an initial position. Note that that white pixels represent padded pixels, as to avoid shrinking of the image if to be done without padding. The filter then does a computation on the region and returns a value of that region as a new pixel value (green). The filter then slides a pixel-width to the right (convolves) and repeats the process until it has convolved the whole image.

In this example, the layer is shown to only apply one filter. Practically, more than one filter may be applied in a single layer. The shape and size of the filter can be manipulated to produce different filter effects to for various object detections. With every convolutional layer, the layers build up in complexity. This is how, as stated before, patterns are detected. From simple lines and curves, building up to edges and corners, developing into shapes and structures, reaching a point where minor details can be detected such as eyes, tails, nose, and then finally, a whole object, such as a dog or a cat.

When an image is large, there is a high change of overfitting due to the amount of details in the image, causing the model to fail to generalize enough. One way to approach this is by reducing the size of the image while trying to maintain as much information as possible. In CNN, this method is called pooling. Pooling functions similarly to how convolution with filters do in CNN. The difference in operations is that pooling is used specifically to reduce the size of the image, and not for detecting patterns. Reducing the size of the image may sound counter-intuitive as it means loss of data. However, using pooling methods such as max-pooling, highly activated pixels are retained, thus aiding in performance of the model.

To explain pooling, a 2x2 max-pooling with a stride of 2 pixels will be illustrated on a 4x4 matrix with values.

The pooling filter begins in the first region and computers the max value. In the first 2x2 region, the max value is 9. The filter, with a stride of two, then moves two pixels to the right and computer the regions highest value, which in this case is 8. Since it has reached the end of the frame, the filter, with a stride of two, moves 2 pixels downwards, and towards the left-most side of the image matrix again, and repeats the process. Once the filter passes over the whole image, the image matrix with max values attained, with respect to the regions the filter was able to traversed is returned. Hence, a smaller imager is returned with the max values of each region.

There are many other pooling methods that can be used for different results. Average pooling is another pooling example. In average pooling, the average value of the region within the filter is returned in the new image matrix.

4.2 Eye Aspect Ratio (EAR)

As the name implies, EAR is a measure used on the eye to determine how open or closed the eye is, based on predetermined landmarks. A common algorithm used for this purpose includes dlib (dLib C++ Library, 2020). The first step of the process would be to map key points, or landmarks onto the face in question. Depending on the algorithm and its complexity, the number of landmarks may vary. In terms of the dlib library, there are 68 landmarks marked for a human face.

From studying the above representation of the landmarks, we can see that points 37 to 42 and points 43 to 48 represent landmarks plotted for the right and left human eye respectively. With these points, the next step is to calculate the ratio of openness of the eyes by applying a ratio for the height of each eye and the length of each eye. For each eye, each landmark corresponds to a point in the formula, such as;

The formula for calculating the EAR value for each eye are as follows;

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

When plotted onto a graph, we see the values of EAR correlate to the behaviour of human eyes when blinking.

With this idea in mind, we can determine the state of the eye based on the EAR measure recorded by comparing it to a pre-set constant, where any value above it is considered as the eye to be open, and any value below it is considered as the eye to be closed

Chapter 5: Data Pre-processing

5.1 Dataset

The dataset that will be used in this study for model training is the MRL Eye Dataset (MRL Eye Dataset, 2021). It in itself a large-scale dataset filled with human eye images. The dataset was created as the team behind it has been involved in solving tasks in the behavior of driver behavior. These tasks do not exclude eye detection, gaze estimation and eye-blinking frequency.

5.2 Data Collection

The dataset itself contains 84898 images of eyes. According to the website, the images in the dataset were captured through infrared means, in both low and high resolution. The eye detector used to obtain the images of the eye were based on a histogram of oriented gradients (HOG) combined with an SVM classifier. The images were also captured in various lighting conditions as well as taken by different devices. The data was collected based on 37 different people, 34 of which were men and the remaining 4 were women. The participants included people who wore and did not wear glasses. The dataset includes images of both open and closed eyes of approximately equal amounts, which prevents oversampling issues. The images also varied in terms of quality of images, which was based on the amount of reflection occurred in the image.

5.3 Data Description

The images produced from this collection were annotated according to the following properties, in the specific order;

- Subject ID
- Image ID
- Gender [0 - man, 1 - woman]
- Glasses [0 - no, 1 - yes]
- Eye state [0 - closed, 1 - open]
- Reflections [0 - none, 1 - small, 2 - big]
- Lighting conditions [0 - bad, 1 - good]
- Sensor ID [01 - RealSense, 02 - IDS, 03 - Aptina]

An example of the annotations is as shown below;

5.4 Data Pre-processing

The data has to be arranged and cleaned if possible before it could be fed into a classifier. The data was primarily cleaned using Python. First the data had to be split into two categories; open and closed eyes. This was done by manually sifting through the folders in the dataset and dividing them accordingly. The reason this was done manually was so that imperfections such as wrongly detected facial parts could be filtered and removed. The two folders, once segregated, each contained an approximate of 42000 samples, which negates the issue of oversampling.

Next the data had to be further pre-processed using Python. The data was first loaded into a 2-dimensional array. The first column stored the images (attributes), which had been saved in grayscale and resized to 50x50 for consistency. The second column stored a number which corresponded to their state (0=Open,1=Close). This was to be the base structure of the dataset for this study. At this stage, the data contained 84606 samples. The images were also normalized by dividing them by a 255.0 to make the range of values for the pixels between 0 to 1. This change does not remove much data from the image. It does however boost the accuracy of the model during training from approximately 69% to approximately 95%.

Finally, the data was shuffled as to avoid overtraining for a specific class and split into its "X" and "y" components. The two arrays were then "pickled" using the Python library "pickle" which saves the data to be loaded once again for model training purposes.

Chapter 6: Experiments and Results

6.1 Evaluation

Due to the lack of standard model evaluation techniques in drowsiness detection, tests have to be manually created based on the type and performance of created models. In this research, the implementations will be evaluated based on their performance in detecting driver drowsiness and the data will be cross-validated. One primary method that will be used as an evaluation method is to analyze the detection rates of the implementations, which is inspired by the works of (Kong, et al., 2014). In this evaluation method, the data was divided into 3 different categories for target detection, namely for the human face, open-eye state, and closed-eye state. The values that will be recorded here will be the number of frames for each of the evaluation criteria. The testing will be done by recording the number of correctly detected frames and the number of missed detection of frames. Together with the total number of target frames for each category, the detection rate for each category can be calculated and analyzed. The categories will be revised to match the purpose of each implementation in this research project. This method can also be used to form a confusion matrix, a set of numbers that display the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). With these values, other metrics can be evaluated such as the Accuracy, Precision and Recall metrics. The Accuracy metric is simply the ratio of correct predictions made. The Precision metric displays the portion of TP among positive predictions. It describes the ratio of correct positive predictions among positive predictions made by the model. The

Recall metric displays the ratio of TP to the sum of TP and FN. Since TP and FN are generally positive instances, recall can also be understood as the ratio of correct positive predictions among positive instances. The formulas of these metrics are as follows;

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Another method that could be applied for evaluation is by preparing short clips of simulated driving to test the implementations on. The implementations will be used to classify the clips into their respective categories, based on the test. The clips will then be manually analyzed for correct detections and false detections, which will enable us to create a confusion matrix for the implementations as well the subsequent derived evaluation metrics described above.

6.2 Experiments

Based on the knowledge from the previous studies, implementations of the driver drowsiness detector can be built. In this study, the aim of the implementation is not to create a better system, but to test the capabilities of a driver drowsiness detection system with various parameters. Therefore, the proposed experiments that follow are intended to reveal in greater detail the characteristics of a drowsiness detection system

6.2.1 Experiment 1

In this experiment, the two applications of a blink detector will be tested against each other to understand which has a better efficiency at detecting blinks. The test will be carried out on a single test video which contains a predetermined set of blinks. The program will then be run on the video to count the number of blinks detected by the applications. The program then outputs the same video with the progress of blink counting overlaid on it, for manual reviewing. The output video is then analyzed, and a confusion matrix is constructed from the observations. Due to the actual "No Blink" state being infinite, correct detections of the class were unable to be recorded, hence the accuracy measure is unable to be calculated. Below are the results for the two applications;

6.2.2 Experiment 2

A similar experiment to experiment previous experiment, this experiment focused on observing the performance of the applications of the blink detectors when the camera was placed at a different angle. As results from *Experiment 1* were obtained at a higher placement of the camera, this experiment was repeated by placing the camera at a lower position so it has a different view of the driver. This view may be a common view as it represents a camera being placed on a vehicle's dashboard.

6.2.3 Experiment 3

The next experiment was conducted on the CNN blink detector model to better understand its relationship to the size of the data. An experiment was set up using the prior set up for the CNN blink detection model. The model was then looped to train with a set number of samples, increasing by 10000 with each iteration. At each iteration, the model

Below is a visualization of the data when the accuracy and the training set size are plotted against each other. The experiment was then reconducted with the train set size representing a logarithmic. The procedures that followed were the same. Below are the results of the experiments;

6.2.4 Experiment 4

After conducting various experiments, both blink detectors were applied in creating the prototypes. The detections were both based on how long the detected eyes were closed for. A point system was then set up to gauge the "drowsiness" of the individual. Points were added and deducted for closed and open-eyed states respectively. When the points reached a certain threshold, the program will trigger an alarm and state that the driver is drowsy. The basic structure of the program can be visualized as below;

The application of the Eye State Evaluation and Scoring step are almost similar for both CNN and EAR application, except for some changes in terms of the evaluation. A flowchart to describe both systems are shown below;

6.3 Discussion

In *Experiment 1*, the two applications of a blink detector were tested on a test video to observe its performance. By comparing the precision measures of both applications, we can see that the CNN blink detector has a higher precision at 0.8 (80%) compared to its EAR counterpart at 0.7 (70%). This shows that the CNN blink detector is able to detect actual states of blinking approximately 10% better than the EAR blink detector. Consequently, in terms of recall, the EAR blink detector has a higher value at 0.808 (80.8%) whereas the CNN blink detector has a lower value at 0.727 (72.7%).

In *Experiment 2*, the two applications of a blink detector are tested at different camera angles. From the results, it is clear the both applications of the blink detectors performed better at elevated angles, compared to that of a lower angle, such as one representing the angle of a camera placed on a dashboard. From this experiment, we see the precision values of the CNN application drop from 0.8 (80%) to 0.333 (33.3%), as well as for the EAR application, where the values drop from 0.7 (70%) to 0.4 (40%). This suggest that the current calibrations for the blink detectors, and subsequently, the drowsiness detector that applies it, would perform better at elevated angles, such as on the sun visor. However, when comparing performances on lowered angles, the EAR application performs better with a precision approximately 7% better than the CNN application. This could suggest that a dashboard angled blink detector and subsequently, its drowsiness detector, could be created by opting for an EAR approach instead of a CNN approach.

In *Experiment 3*, varying dataset sizes were tested with the CNN blink detector model to observe a minimal dataset size for training a blink detector with the current calibrations. From the first part of this experiment, the results showed a minor increase to the accuracy of the models as the dataset size increased from 10000 samples to 80000 samples. There were more observable results with the loss metric, seeing a steeper decrease in loss as the number of samples increased, albeit an outlier in the data that occurred when 40000 samples were used. The experiment was repeated in the second part, this time using a logarithmic scale of values. This was done to identify an "elbow" in the graph which could signify a point where the increase in accuracy of is occurring at a reduced rate. From my findings, it is observed that the increase in data begins to slow down at approximately 8000 samples in the data. This is also reflected in observing the loss of the model, where the drastic decrease in value slows at around 8000 samples in the data. From this experiment, the minimal size of data that should be used should be 8000 samples for better performance. More data can be used; however, the performance of the model would not be affected by much. To compensate for a lack of data beyond 8000 samples, one can improve the performance of the model by conducting parameter tuning on the model itself to increase its performance.

Experiment 4 involved the accumulation of current knowledge to build a drowsiness detector. The main flow was inspired by (Rosebrock, 2020). From the two builds, there were two main types components that could be calibrated to change the performance of the drowsiness detectors. They were mainly the eye state thresholds and the score thresholds. The eye-state threshold serves to differentiate the states of the human eye between opened and closed states. In the CNN approach, this is done by training a model to classify the eyes as opened or closed. In the EAR approach however, the eye-state threshold is represented as a constant. In the current implementation, the EAR threshold is set to 0.23. Despite it being represented as a constant in this application of a drowsiness detector, it should be noted that this value is subjected to change based on the individual, as different people will have different EAR constants that draw a line between their opened and closed states of the eye. Individuals whom have naturally large eyes and individuals who squint often will require different EAR constants. The score threshold is present in both applications of the drowsiness detector. It represents the score point which will be used to declare whether the individual is drowsy or not. When the eyes are opened, the score is reduced by one each frame until 0. When the eyes are closed, the score begin to add up until the score threshold is surpassed, at which, an alarm will sound and the screen will notify of drowsiness. Once again, in this implementation, the score threshold is set to a constant of 15 (15 frames of closed eyes in a short amount of time). This value can vary as drowsiness is not always defined by a pre-determined amount of time that the eyes are closed. The time of closed eyes to indicate drowsiness can also vary between individuals with different characteristics. The value may also change with different hardware as devices that record in higher framerates will require a higher threshold value.

6.4 Gantt Charts

Below is planned Gantt Charts which was used for the two semesters to conduct this study.

Chapter 7: Conclusion

7.1 Summary

In completion of this study, the idea of a drowsiness detector is understood with greater clarity. The technology no longer seems as a figment of science fiction, but more of a practical implementation that can potentially save lives, as well as provide a deeper understanding of human behavior

By conducting a literature review on the many types of drowsiness detectors as well as their modes of implementations, their individual characteristics could be revealed and their benefits and disadvantages are clearly understood. Other than understanding how the implementations worked, the process of the literature review also revealed a learning path to constructing an implementation. By learning the required components of a drowsiness detector such as the facial recognition aspect and models such as a Convolutional Neural Network (CNN), it paved a path to constructing the two basic forms of the drowsiness detector for learning purposes. Once an understanding of the concepts had been solidified, a series of experiments was planned to further test the characteristics and limitations of a drowsiness detector. Experiments were executed which tested several aspects of the detector and its performance was observed and evaluated through known metrics.

The objectives that were set out for this study were also achieved. By conducting a background study on the current achievements of drowsiness detectors as well as studying their applications, characteristics, and limitations, relevant information was gained in order to construct a prototype form of a drowsiness detector that can be potentially applied with further development. The facial features that was focused in this study was the eyes of the driver, which was studied in forms of classification methods. By understanding these methods, the core function of the drowsiness detector which was a blink detector, was created and analyzed.

7.2 Limitations

The main limitations of the applications of driver drowsiness detectors are that they are not dynamic. Both applications function-based on a rudimentary idea of drowsiness based on length of blinks. Although it attends to the base idea of monitoring the length of blinks to determine signs of drowsiness, the application of these idea is considered a static approach as it has a pre-determined set of constants for its scoring system for the drowsiness detection, such as specified number of frames (length of blink) and EAR threshold in terms of the EAR. These constants could limit the capability of the prototypes to perform in various different scenarios under different influences, such as different types of drivers with different resting EAR.

The applications are also limited in facial behavioral factors. The current study has focused solely on the human eye to determine if the driver is drowsy. In the event that the driver's eyes are not accessible to the application (i.e. glare effect, driver wearing sunglasses), the application could be rendered useless. The dependency of the eye of the driver is too heavy in this form of the prototype, making it one of its biggest limitations.

Another limitation of this study is a proper evaluation method to evaluate the performance of the application. For instance, the performance of the two blink detectors were evaluated manually on a predetermined set of 30 blinks, which is considered a small sample size. The evaluation process was also flawed where if a mistake were to be made in the process of classifying the detections, the evaluation for the video had to be restarted.

To evaluate a greater sample size would require a greater amount of effort and lesser opportunities of human error.

7.3 Future Work

Current works on the driver drowsiness detector are not near groundbreaking work. There is still a plethora of improvements that can be made to further solidify this idea of a safety measure. For starters, the same set of studies could be made for other parts of the human face to study for relevance to drowsiness. For example, the mouth can be observed for signs of yawning which could contribute to signs of drowsiness. To take it a step further, higher performing applications could study major facial muscle points, such as the cheeks or the forehead, to study its relaxation and contraction rates in order to detect drowsiness.

The application can also be made to perform more dynamically by removing its reliance on preset constants. As stated in the limitations, the current prototypes have a heavy reliance on certain preset constants to detect drowsiness such as the number of frames and EAR threshold for the EAR application. These constants should be replaced by models or equations that are able to generate the required values to allow the application to perform with different individuals under different circumstances.

Finally, more work could be conducted on the image retrieval for the facial behavioral factors on the human face. Applying different types of systems for capturing the data may improve the performance of the applications, as well as open more room to expand on the ideas for drowsiness detection. To be able to work on different types of data such as new key points on the human face could provide a greater understanding of the behavior of humans, thus improving our ability to create applications to detect drowsiness.

FYP2

ORIGINALITY REPORT

3%	2%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Advances in Visual Computing", Springer Science and Business Media LLC, 2018 Publication	<1%
2	trepo.tuni.fi Internet Source	<1%
3	www.swov.nl Internet Source	<1%
4	SpringerBriefs in Computer Science, 2014. Publication	<1%
5	Dikmen, I.. "A case-based decision support tool for bid mark-up estimation of international construction projects", Automation in Construction, 200711 Publication	<1%
6	mafiadoc.com Internet Source	<1%
7	Hera Khan, Ayush Srivastav, Amit Kumar Mishra. "Use of Classification Algorithms in Health Care", Emerald, 2020	<1%

Publication

8	digitalcommons.morris.umn.edu Internet Source	<1%
9	www.scielo.mec.pt Internet Source	<1%
10	mrl.cs.vsb.cz Internet Source	<1%
11	www.thestar.com.my Internet Source	<1%
12	www.guru99.com Internet Source	<1%
13	pubs.rsc.org Internet Source	<1%
14	aaltodoc.aalto.fi Internet Source	<1%
15	av.dfki.de Internet Source	<1%
16	towardsdatascience.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

FYP2

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37