# Experiments and Results

## 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;

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.

## 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

### 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;

Table 4 EAR Blink Detection Confusion Matrix

Table 3 CNN Blink Detection Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink | 24 | 9 |
| No  Blink | 6 | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink | 21 | 5 |
| No  Blink | 9 | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink Detected | 24 | 9 |
| Blink not Detected | 6 | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink Detected | 12 | 0 |
| Blink not Detected | 18 | - |

Precision :0.800

Recall :0.727

Precision :0.700

Recall :0.808

### 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.

Table 3 CNN Blink Detection Upper View Confusion Matrix

Table 4 CNN Blink Detection Lower View Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink Detected | 21 | 5 |
| Blink not Detected | 9 | - |

Precision :0.333

Recall :0.909

Precision :0.800

Recall :0.727

|  |  |  |  |
| --- | --- | --- | --- |
| Possible Decision Distribution | | Actual State | |
| Blink | No Blink |
| Result of Detection | Blink Detected | 10 | 1 |
| Blink not Detected | 20 | - |

Table 5 EAR Blink Detection Upper View Confusion Matrix

Table 6 EAR Blink Detection Lower View Confusion Matrix

Precision :0.400

Recall :1.000

Precision :0.700

Recall :0.808

### 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 was tested with 4806 samples that were not included in the dataset used for training. The results of the data are as shown below;

Table 7 Accuracy and loss measures for varying dataset sizes pt. 1

|  |  |  |
| --- | --- | --- |
| **Dataset Size** | **Loss** | **Accuracy** |
| 10000 | 0.132861 | 0.953184 |
| 20000 | 0.114698 | 0.958177 |
| 30000 | 0.07985 | 0.971702 |
| 40000 | 0.150092 | 0.949022 |
| 50000 | 0.072917 | 0.974615 |
| 60000 | 0.067462 | 0.975864 |
| 70000 | 0.063404 | 0.978568 |
| 80000 | 0.047259 | 0.98377 |

Below is a visualization of the data when the accuracy and the training set size are plotted against each other.

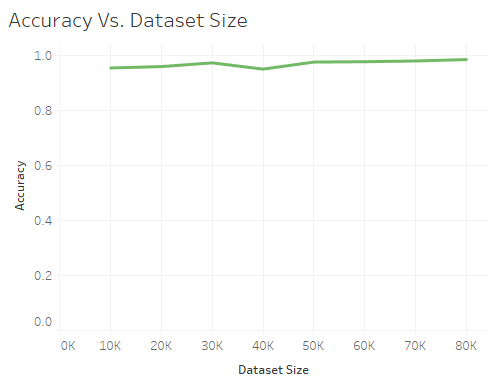
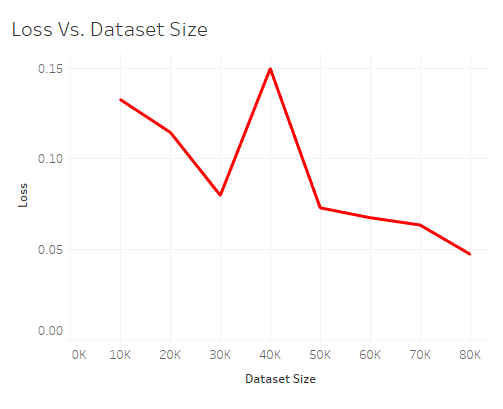
The experiment was then reconducted with the train set size representing a logarithmic. The procedures that followed were the same. Below are the results of the experiments;

Figure 6.1 Loss Vs. Dataset Size pt.1

Figure 6.2 Accuracy Vs. Dataset Size pt.1

|  |  |  |
| --- | --- | --- |
| **Dataset Size** | **Loss** | **Accuracy** |
| 80 | 0.678238 | 0.532043 |
| 800 | 0.273736 | 0.896171 |
| 8000 | 0.145054 | 0.950895 |
| 80000 | 0.059834 | 0.979609 |

Table 8 Accuracy and loss measures for varying dataset sizes pt. 2

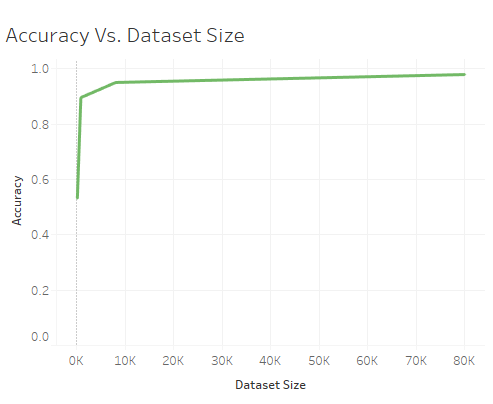
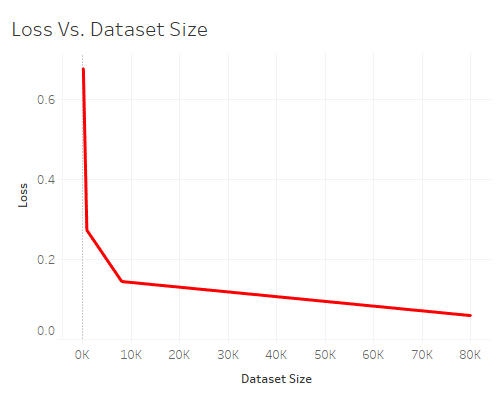
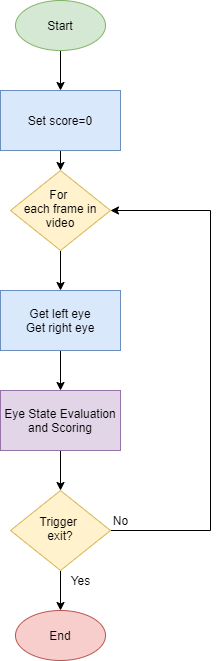


Figure 6.3 Loss Vs. Dataset Size pt.2

Figure 6.4 Accuracy Vs. Dataset Size pt.2

### 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;

Figure 6.5 General Flow of applied drowsiness detection systems

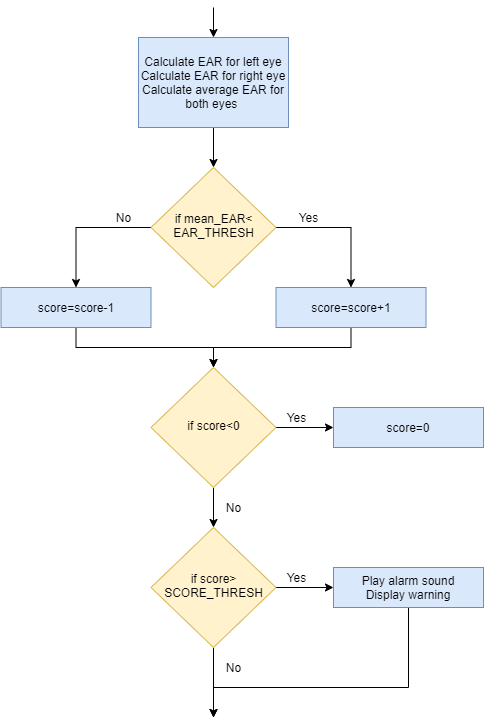
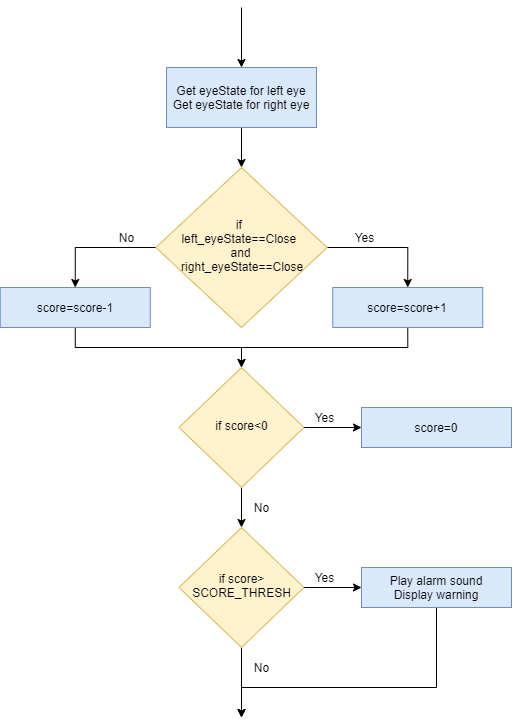


Figure 6.6 Eye State Evaluation and Scoring step for CNN Drowsiness Detection

Figure 6.7 Eye State Evaluation and Scoring step for EAR Drowsiness Detection

## 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.

## Gantt Chart

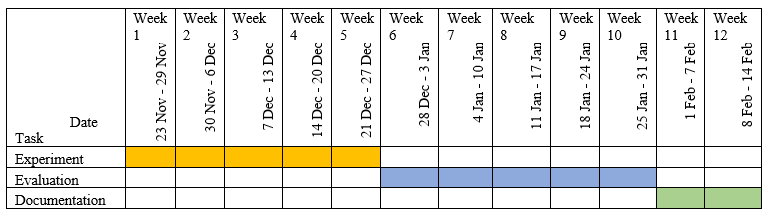
Below is the planned structure for the second part of the research in the upcoming trimester. The plan was drafted out based on the assumption that the project duration would be from Week 1 to Week 12 of Trimester 2,2020/2021. The tentative plan is also based on the idea of spending equal amounts of time for both experimentation and evaluation, and allocating a remainder of two weeks for the documentation.

Table 9 Tentative Gantt Chart for the upcoming half