Team 56 - Project Final Report

SleepCatcher

Team Member	Student Number
Ajeya Madhava Rao	1006098287
Alastair Sim	1006460287
Louis Liu	1005202424
Yi Lian	1005709333

Word Count: 2421 **Penalty: 0**

GitHub Link: https://github.com/splouisliu/driver-drowsiness-detection

1. Introduction

On average, 16 thousand car accidents occur every year in Canada, of which more than 2500 resulted in deaths, with fatigue being the main cause [1]. Drowsy driving can be as dangerous as drinking or texting while driving, yet around 60% of drivers have gotten behind the wheel feeling tired [2]. There are many factors that contribute to sleep-impaired driving, leading to injuries, deaths, and financial losses [3].

In order to combat this issue, a drowsiness detection system can be implemented to alert drivers before an accident occurs. A simple car-mounted camera would capture the driver's face and pass real-time video data into our model to obtain continuous predictions on the driver's drowsiness. Once it predicts *drowsy*, proper action such as alerting the driver can be taken.

The scope of our project can be summarized as (see Figure 2.1):

* **Image Extraction:** Extract image sequences from a real-time video of a driver[1]
1. **Eye Detection:** Identify and crop the left eye
2. **Eye Classification:** Classify whether the eye image is open or closed
3. **Drowsiness Classification:** Calculate the percentage eye closure (PERCLOS) over a sequence of video frames and make a decision using a reference threshold

Our project emphasizes on stage 2, an inherent image classification problem. Deep learning with CNN would be the most suitable approach because it can extract features, pick up spatial information, and enable weight sharing. It can effectively learn from an eye database and accurately predict eye states.

[1] The image extraction was implemented via OpenCV, but not discussed in this report. See GitHub code for more details.

2. Illustration / Figure

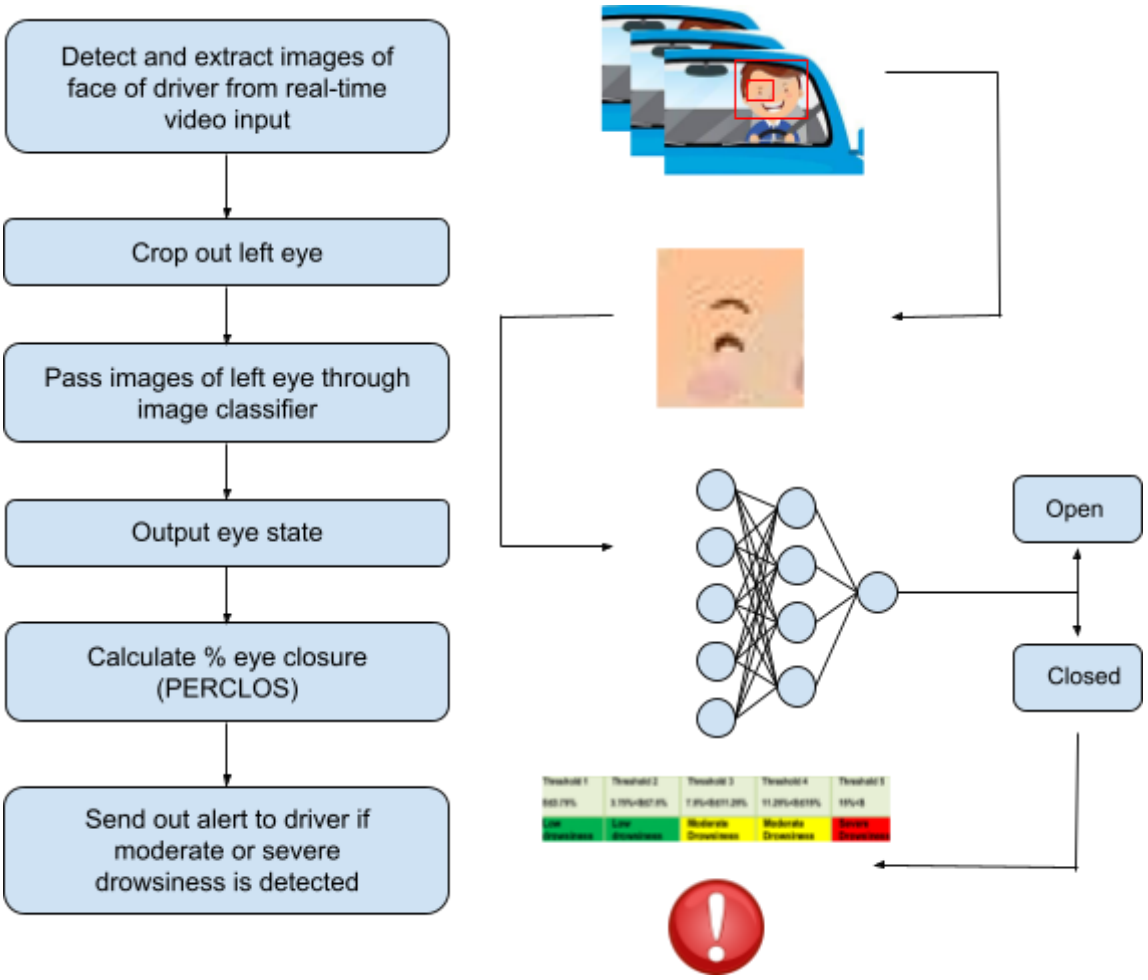

Figure 2.1 General overview of project.

3. Background & Related Work

With fatigue and sleep deprivation being a main cause of serious motor vehicle accidents, an extensive amount of research has been done on prevention of driver drowsiness. A potential solution developed by the Bosch Group is an algorithm which monitors the steering movements and advises drivers to take breaks [4]. It uses a steering angle sensor which sends approximately 70 signals that are evaluated by the algorithm. Others have also researched into developing a system using convolutional neural networks (CNN) which detect the states of the driver's eyes and mouth from images, producing an accuracy of 93.623% and a sensitivity rate of 93.643% [5]. The percentage of eyelid closure over time and mouth opening degree are two parameters used in determining the fatigue level of the driver. A team at UT Austin has done this by collecting drowsy and alert video data from 60 participants [6]. The data was processed using OpenCV and fed into their ML algorithm.

4. Data Processing

Our project utilized three different datasets: MRL [7], DROZY [8], and UTA [9].

The MRL dataset used for training the eye classifier containing grayscale images of eyes from 37 different subjects: 34 males and 3 females. The data consists of various conditions describing whether the subject had glasses, eye state, and lighting condition. It was recorded using 3 different sensors which generated a range of image resolutions consisting of 640 x 480, 1280 x 1024, and 752 x 480 [7]. Only the eye state property was used as ground-truth, and all examples were used.

The DROZY dataset consists of 14 young subjects: 3 males and 11 females. The subjects performed 3 different 10-minute psychomotor vigilance tests (PVT) at 3 different stages of drowsiness. The database contains time-synchronized data consisting of KSS scores ranging from 1-9 and Kinect v2 sensor videos which are grayscale and 512 x 424 – manually and automatically obtained from facial landmarks [8]. This dataset was used to compare the differences between models and image recognition software.

The UTA dataset contains 180 RGB videos of 60 participants with 3 different KSS scores. The participants consist of 51 males and 9 females from different ethnicities of 10 Caucasian, 5 non-white Hispanic, 30 Indo-Aryan and Dravidian, 8 Middle Eastern, and 7 East Asian. Their ages ranged from 20 to 59 years old. The videos were taken from different angles in real-life environments such as in a car [9]. This dataset was used to evaluate our model's capabilities in real-life applications.

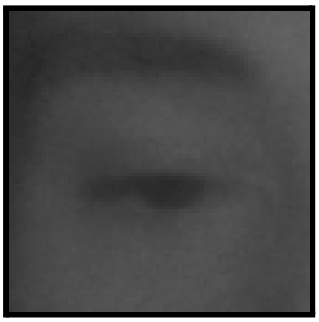

Figure 4.1 MRL dataset.[7]

Figure 4.2 DROZY dataset.[8]

Figure 4.3 UTA dataset.[9]

To load the MRL dataset, a custom Dataset class was created inheriting torch.utils.data. Since the MRL data contained many other property labels that we disregarded, the data was iterated over to sort the images and relabel them to only include the eye state as the label. The data was split using a 60-20-20 split for training, validation, and test sets respectively.

Data augmentation was used to generate more training data and increase the performance of the model in the wild. The following transforms were applied using a for loop to each training image:

- Gaussian blur with kernel size 3
- Random rotation of up to 30 degrees
- Random horizontal flip with 25% probability of occurrence
- Random resized crop to a final tensor of 1x32x32

Using these transforms allowed our model to account for noise and different eye orientations. Applying data augmentation generated 2911972 images from the 59428 images in the training set. A simple transform was also done to resize the validation and testing images to 32x32 pixels.

5. Architecture

The proposed model consists of three stages: eye detection, eye classification, and drowsiness classification.

5.1 Eye Detection

In the eye detection stage, the model first locates the subject's face within an image using an object detection algorithm. This is the most computationally expensive process in our pipeline, and in order to design for real-time processing, we set our model requirement to process videos up to 30 frames per second. After evaluating four different pre-trained face detection models in terms of speed and performance (Appendix A), the chosen method is the MobileNet Single-Shot Detector (SSD)[10]. If multiple faces are detected within an image, the one with the highest confidence would be selected.

Upon locating the face, the model uses an ERT-based facial landmark[11] to identify 68-points, of which a square bounding box is cropped around the left eye including the eyebrows (Figure 5.1). This project assumes that the subject's left and right eyes blink uniformly, therefore, only one eye is required for analysis.

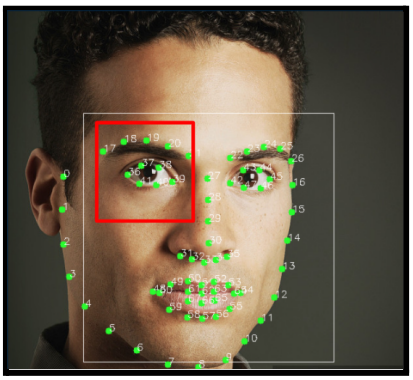

Figure 5.1 Illustration of eye detection stage.[12]

5.2 Eye Classification

In the eye classification stage, the cropped image of the left eye is fed into a model to determine its eye state: open or closed. An eye that is more than 80% closed is classified as closed. The selected approach is a CNN which outperforms a simple artificial neural network (ANN) baseline model, which were both trained and optimized on the MRL dataset.

5.2.1 Baseline Model

In order to compare results from our CNN model, an ANN baseline model was built. The ANN takes a flattened tensor of size 3x24x24 as the input and has three fully connected hidden layers along with ReLU activation functions to recognize complex patterns of the eye, giving an output size of 2. The Cross Entropy Loss function and the Adam optimizer were used during the training process. Although the ANN has many deficiencies compared to the CNN, the model was still able to produce an accuracy of 86.61%, providing a good baseline comparison for the CNN.

5.2.2 CNN Model

Our final model, DataAug3, was trained on the augmented training set with Adam optimizer and Cross Entropy Loss. The architecture consisted of 2 convolutional layers using kernel size 3, each followed by respective max pooling layers using kernel size 2 and stride 2. The first convolutional layer took 1 input channel and 16 output channels. The output was max pooled and sent into the next convolutional layer which outputs 32 channels. The final output tensor size was 1152 which was sent into the linear classification section. The classification part of the architecture consisted of 2 linear layers. The 1152 input was fed in through 256 hidden neurons and output with a size of 2.

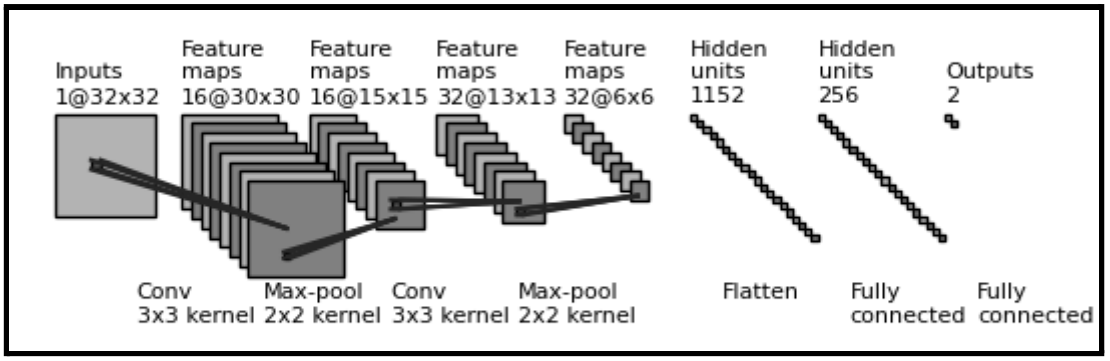

Figure 5.2 CNN Architecture (see Appendix B)

The final model was trained using the following hyperparameters:

Table 5.1 Hyperparameters of Best Model.

Batch Size	**Learning Rate**	**Epochs**
64	0.001	20

5.3 Drowsiness Classification

Once the eye state has been outputted by the CNN model, the drowsiness level of the driver is determined using PERCLOS (percentage of eyelid closure over the pupil over time):

$$PERCLOS = \frac{Number\ of\ Closed\ Eye\ Frames}{Total\ Number\ of\ Frames\ over\ 1\ minute}$$

The calculated value is compared to the PERCLOS threshold values to establish the drowsiness level.

Threshold 1	Threshold 2	Threshold 3	Threshold 4	Threshold 5
S≤3.75%	3.75%<S≤7.5%	7.5%<S≤11.25%	11.25%<S≤15%	15%<S
Low drowsiness	Low drowsiness	Moderate Drowsiness	Moderate Drowsiness	Severe Drowsiness

Figure 5.3 PERCLOS threshold values.[13]

6. Evaluate Model On New Data

The CNN eye classifier was trained on the MRL eye dataset which contains only grayscale images of open and closed eyes of different subjects, and attained very promising test results. However, it is only one piece of the puzzle as our goal is to predict the drowsiness level of the person in a real-time video.

An overall evaluation of the model encompassing all three stages was conducted. The testing dataset consists of DROZY and UTA datasets with videos of subjects in controlled and real-world settings, respectively, never seen by the CNN classifier. It also contains self-assessed drowsiness scores using the Karolinska Sleepiness Scale (KSS), a 9-point scale rated subjectively (Figure 6.1).

The testing procedure starts with feeding a video into OpenCV to continuously extract image frames. These image sequences are preprocessed using histogram equalization and converted to grayscale 1x32x32 tensors. They are inputted into the proposed model which then outputs a rolling PERCLOS score calculated over the past minute. The PERCLOS score [0–100%] is normalized to KSS scale [1–9] using a three-part mapping function (Figure 6.2) divided proportionally based on the drowsiness category thresholds in Figure 5.3. The predictions are then compared to the subjects' self-assessed scores. This can be seen on the top left region in Figure 6.3 as the "Self Assessment: 4 – Low Drowsiness" and "Prediction: 3 – Low Drowsiness".

Rating	Verbal descriptions
1	Extremely alert
2	Very alert
3	Alert
4	Fairly alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep alert
8	Sleepy, some effort to keep alert
9	Very sleepy, great effort to keep alert, fighting sleep

Figure 6.1 Karolinska Sleepiness Scale [14]

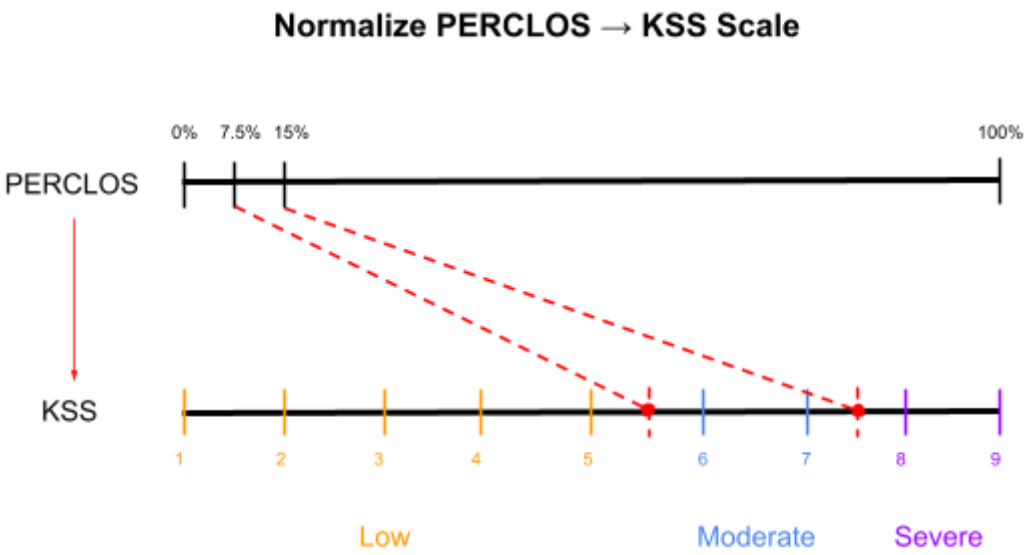

Figure 6.2 Mapping of PERCLOS to KSS scale.

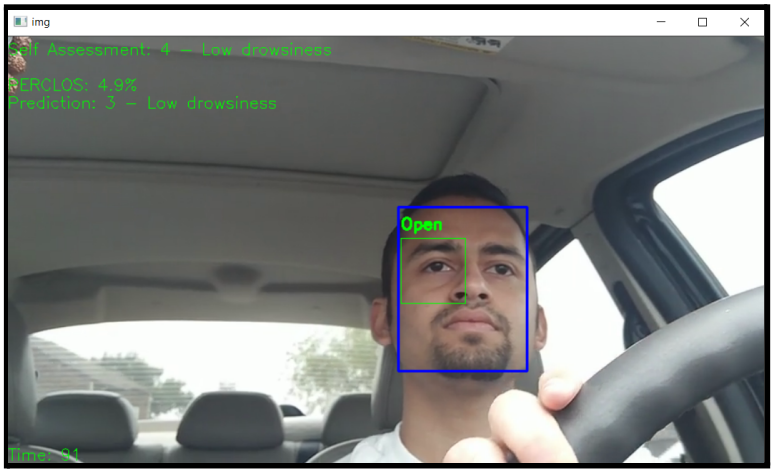

Figure 6.3 SleepCatcher's prediction (top left), 3, is very close to the subject's self assessment.[9]

7. Quantitative Results

7.1 CNN

In order to measure quantitative results for our CNN model, the loss and accuracy training curves were plotted. The final training and validation accuracies were 99.8% and 98.8%, respectively. The model was able to obtain a test accuracy of 98.9% on the MRL test set. A confusion matrix was also created to provide a visual representation of our results.

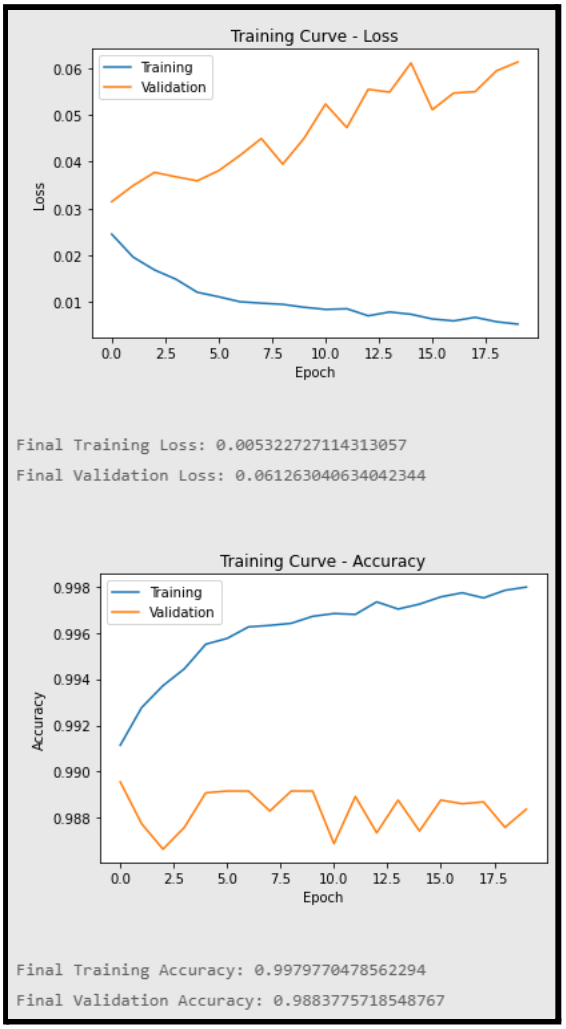

Figure 7.1 Training curves, training and validation accuracies.

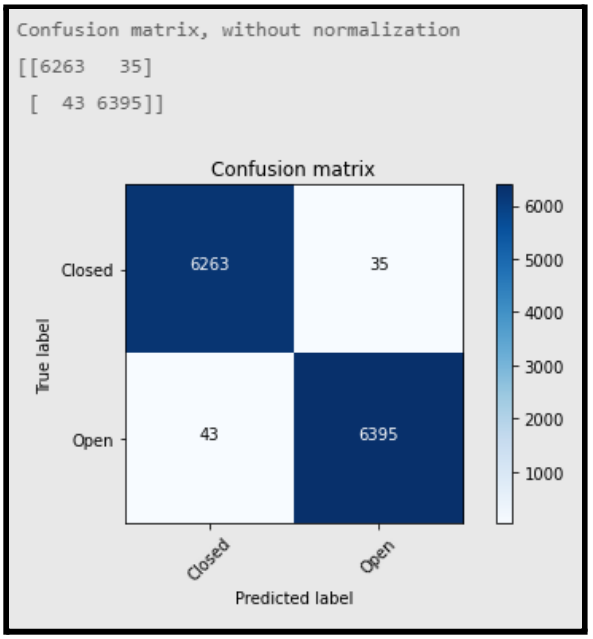

Figure 7.2 Confusion matrix.

Table 7.1 Final Results.

Test Accuracy	Precision	Recall
98.9%	99.4%	99.3%

7.2 Overall Evaluation

As described in Section 6, an overall evaluation of the entire model was conducted on the DROZY dataset. Using the Karolinska Sleepiness Scale, the subjects' self assessed scores were compared to the normalized predictions outputted by the proposed model in Figure 7.3. A wrongful prediction is defined as one that falls under a different drowsiness category (low, moderate, severe) than the subject's self assessment. There are a total of 12 wrongful predictions across all 36 videos, which constitutes **33% prediction error** using the formula below.

$$Total\ Prediction\ Error = \frac{Wrongful\ Predictions}{Total\ Predictions}$$

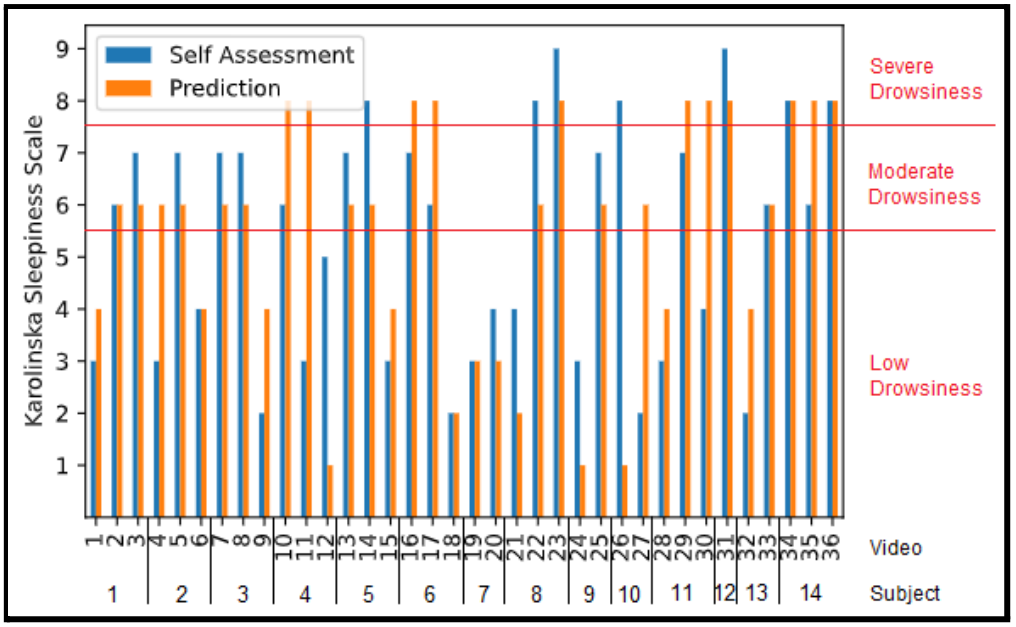

Figure 7.3 Comparison of DROZY subjects' self-assessments and model's predictions

Another metric called the KSS error is defined to represent the absolute deviation of a model's prediction from the subject's self assessment, regardless of the drowsiness categories. The results are shown in Figure 7.4. The average KSS error across all 36 videos is **1.64**.

$$KSS\ Error = |Prediction - Self Assessment|$$

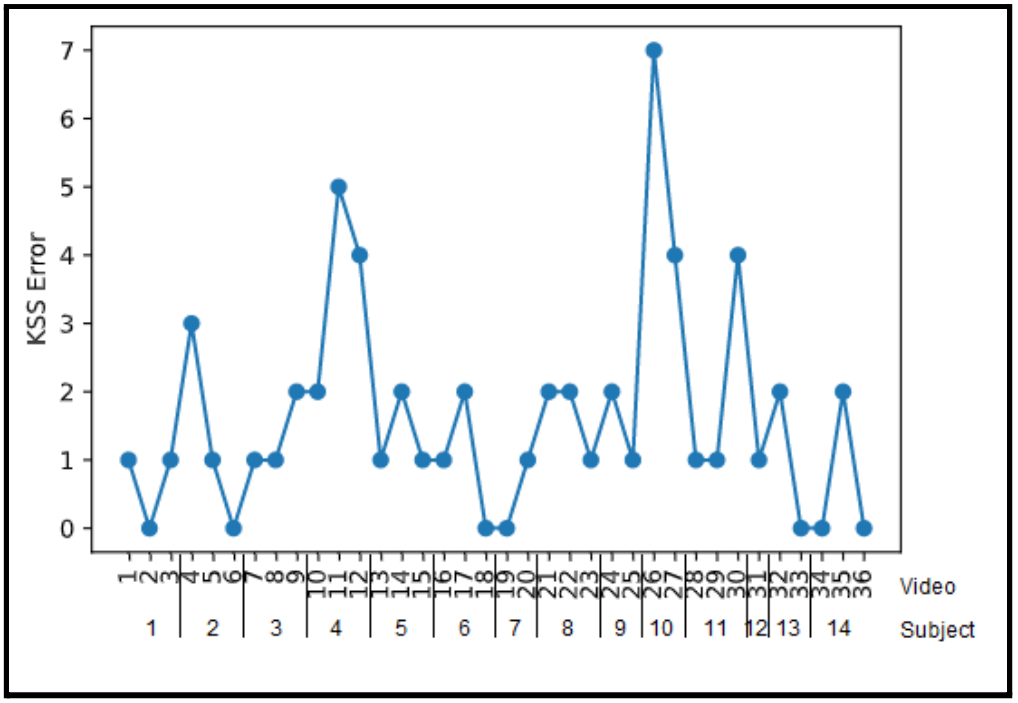

Figure 7.4 KSS Errors of DROZY subjects' self-assessments and model's predictions.

8. Qualitative Results

When evaluated on the DROZY and UTA datasets, the CNN model using the data augmentation performed generally well. On some subjects the model would not work at all (See Figures 8.1–8.6) but on others it would work incredibly well. This could be due to the subject's eye shape not being in the training dataset as the dataset only consisted of 37 subjects.

Figure 8.1 Subject 5 from DROZY showing CLOSED eye being classified as OPEN.

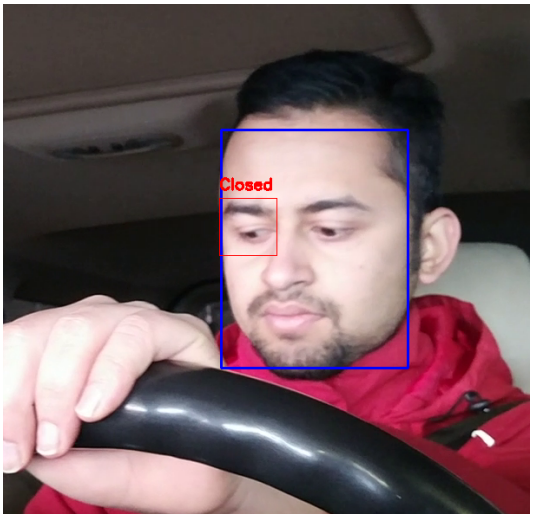

Figure 8.2 Inaccurate prediction due to looking down.

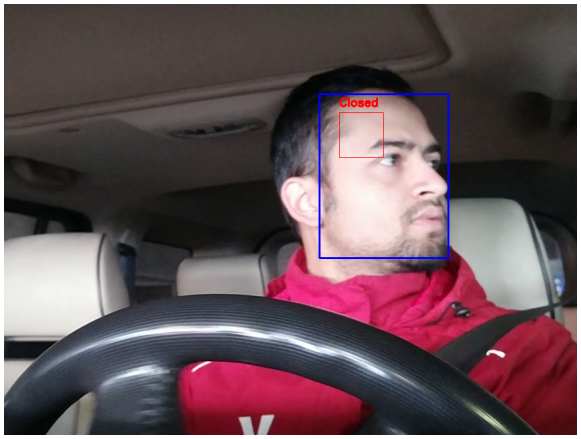

Figure 8.3 Inaccurate prediction from uncommon angle.

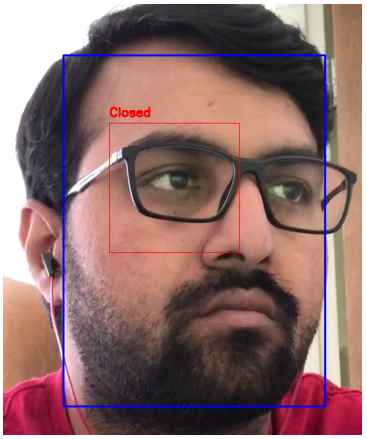

Figure 8.4 Inaccurate prediction due to glasses.

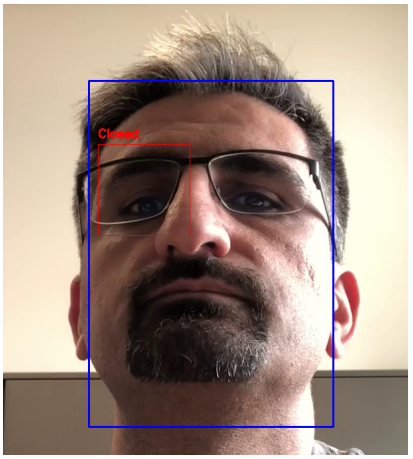

Figure 8.5 Inaccurate prediction due to glasses, uncommon angle, and dark lighting.

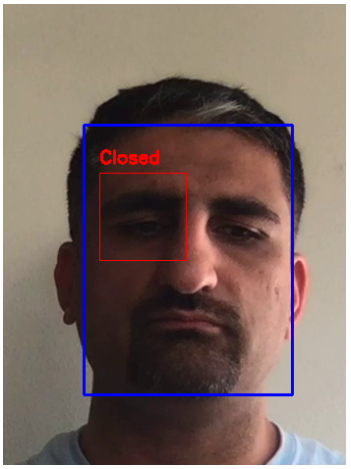

Figure 8.6 Inaccurate prediction due to dark lighting.

The UTA dataset showed that our model would sometimes predict looking down from a certain angle as being closed due to the definition of eye closure being 80% of the eye being covered. The videos also revealed that the model had trouble with glasses.

9. Discussion

Overall, our CNN eye classifier performed very well on the dataset it was trained on, attaining 98.9% training accuracy, 99.4% precision, and 99.3% recall. However, it did not work as well on completely foreign faces in the DROZY and UTA datasets. In our qualitative observations, it was noted that certain subjects were prone to high errors in eye state classification. Some patterns include glasses, bad lighting conditions, and routine half blinks that were not accounted for by our 80% eye-closed definition. There were also cases where no noticeable cause was observed. These errors are likely due to the limitation that the model was trained on only 37 subjects, which is not enough to generalize all eye shapes, patterns, and conditions. Although image augmentation and histogram equalization greatly increased the performance of the model for unseen subjects and extreme lighting conditions, there is still room for improvement. Nonetheless, excellent eye classification predictions were produced on subjects that the model was able to detect.

In the overall model assessment, a high prediction error of 33% was observed. Because this metric is calculated based on correct classification using the three drowsiness categories, it is possible to produce an error when the prediction and self assessment KSS scores are close, but reside across two categories. Additionally, an average KSS error of 1.64 was observed. This indicates that on average, the model prediction is only 1.64 KSS score away from the subject's self assessment, which is very impressive. When looking at Figure 7.4, the KSS errors for most subjects reside around 2; they are drastically high for only a few subjects, namely 4 and 10 with errors up to 5 and 7. Two conclusions can be drawn. First, the high prediction error is likely the result of suboptimal PERCLOS threshold choices. Second, the

model performs very well on subjects that work, which corroborates the previous observation from eye classification.

Another cause for overall mediocre results is eye detection. The ERT-based facial landmark predictor fails to accurately locate the left eye when the face is angled, which results in erroneous eye crops passed to the eye classifier. It is possible to mitigate this problem by utilizing newer state-of-the-art landmark predictors such as PFLD [15] and implementing a smart algorithm that analyzes both eyes instead of one.

All in all, the project was successful. The detection system is able to function on real-time videos from start to finish, and a correlation was found between our predictions and the subjects' self-assessment scores.

10. Ethical Considerations

If our model is to be implemented, we will have to consider both the consensual and non-consensual collection of driver data that our model could be used for. This could allow further development of the model but at the cost of users' privacy. Another consideration could be someone using our model to detect drowsy people as vulnerable targets for crimes such as theft. Furthermore, would the model serve as a distraction to drivers and actually cause an increase in accidents is another consideration.

11. References

[1] Tests.ca, “2020 Driving Statistics: The Ultimate List of Canadian Driving Stats,” Prepare for Your Driving Test in Canada for Free - Tests.ca. [Online]. Available: https://tests.ca/driving-statistics/. [Accessed: 12-Feb-2021].

[2] T. I. Ltd., “The Dangers Falling Asleep While Driving And Drowsy Driving Laws,” Falling Asleep While Driving, Drowsy Driving, Laws, Tickets, FAQS, 22-Feb-2018. [Online]. Available: https://www.thinkinsure.ca/insurance-help-centre/falling-asleep-and-drowsy-driving-laws.html. [Accessed: 12-Feb-2021].

[3] A. Sahayadhas, K. Sundaraj, and M. Murugappan, “Detecting driver drowsiness based on sensors: a review,” Sensors (Basel, Switzerland), 07-Dec-2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3571819/#:~:text=In%20recent%20years%2C%20driver%20drowsiness,driver%20before%20a%20mishap%20happens. [Accessed: 12-Feb-2021].

[4] “Driver drowsiness detection,” Bosch Mobility Solutions. [Online]. Available: https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/driver-drowsiness-detection/. [Accessed: 12-Feb-2021].

[5] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, “Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN,” Computational Intelligence and Neuroscience, 18-Nov-2020. [Online]. Available: https://www.hindawi.com/journals/cin/2020/7251280/. [Accessed: 12-Feb-2021].

[6] G. Zhong, “Drowsiness Detection with Machine Learning,” Medium, 29-Jan-2020. [Online]. Available: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a. [Accessed: 12-Feb-2021].

[7] R. Fusek, “MRL Eye Dataset,” Media Research Lab, 2018. [Online]. Available: http://mrl.cs.vsb.cz/eyedataset [Accessed: 09-Apr-2021].

[8] Q. Massoz, T. Lagohr, C. François, and J. G. Verly, “The ULg multimodality drowsiness database (called DROZY) and examples of use,” 26-May-2016. [Accessed: 09-Apr-2021].

[9] R. Ghoddoosian, M. Galib, and V. Athitsos, “A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection,” UTA-RLDD, 15-Apr-2019. [Accessed: 09-Apr-2021].

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv.org*, 17-Apr-2017. [Online]. Available: https://arxiv.org/abs/1704.04861. [Accessed: 09-Apr-2021].

[11] V. Kazemi, J. Sullivan, “One Millisecond Face Alignment with an Ensemble of Regression Trees,” KTH, Royal Institute of Technology Computer Vision and Active Perception Lab [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kazemi_One_Millisecond_Face_2014_CVPR_paper.pdf [Accessed: 09-Apr-2021].

[12] “Dlib Python Face Landmark Detection,” [Online]. Available: https://line.17qq.com/articles/nmhpndglv.html [Accessed: 09-Apr-2021].
[13] T. P. Nguyen, M. T. Chew, S. N. Demidenko, M. Y. Hossain, F. P. George, F. Neumann, V. Oberhauser, J. Kornmeier, C. François, T. Hoyoux, T. Langohr, J. Verly, Z. Lin, C. Xiao, L. Glass, J. Sun, F. Klewitz, M. Nöhre, M. Bauer-Hohmann, and M. de Zwaan, “TABLE 1 . DROWSINESS LEVELS BASED ON THE PERCLOS THRESHOLDS,” *ResearchGate*, 29-Sep-2020. [Online]. Available: https://www.researchgate.net/figure/DROWSINESS-LEVELS-BASED-ON-THE-PERCLOS-THRESHOLDS_tbl1_283018835. [Accessed: 10-Apr-2021].
[14] A. Sahayadhas, K. Sundaraj, M. M, and R. Palaniappan, “Table 1 Karolinska sleepiness scale (KSS) ,” *ResearchGate*, 08-Dec-2020. [Online]. Available: https://www.researchgate.net/figure/Karolinska-sleepiness-scale-KSS_tbl1_236968173. [Accessed: 10-Apr-2021].
[15] X. Guo, S. Li, J. Yu, J. Zhang, J. Ma, L. Ma, W. Liu, and H. Ling, “PFLD: A Practical Facial Landmark Detector,” *arXiv.org*, 03-Mar-2019. [Online]. Available: https://arxiv.org/abs/1902.10859. [Accessed: 10-Apr-2021].

Appendix A: Testing of Various Face Detection Methods on DROZY Dataset

	Haar Cascades	HoG	SSD	MTCNN
Processing Time	~50 fps	~ 45 fps	~37 fps	~ 14 fps
Performance	*Bad:* - Cannot detect angled faces - Many false positives	*Moderate:* - Small bounding box sometimes cuts into eyes	*Good:* - Accurately detects faces in various conditions	*Good:* - Accurately detects faces in various conditions

Appendix B: CNN Configuration

Layer	Feature Maps	Output Size	Kernel Size	Stride
Input Image	-	$1 \times 32 \times 32$	-	-
Convolutional Layer 1	16	$16 \times 30 \times 30$	3×3	1
Max Pooling	-	$16 \times 15 \times 15$	2×2	2
Convolutional Layer 2	32	$32 \times 13 \times 13$	3×3	1
Max Pooling	-	$32 \times 6 \times 6$	2×2	2
Fully Connected Layer 1	-	256	-	-
Fully Connected Layer 2	-	2	-	-