**Group 6 Project Final Report**

**Project Title**: Driver drowsiness detection using deep learning

**Group Members**: Tianhan Jiang, Peiyun Zhao, David Laditan, David Guo, Tobi Lawal

**Contact Email**: [tianhan.jiang@ucalgary.ca](mailto:tianhan.jiang@ucalgary.ca), [peizhao@ucalgary.ca](mailto:peizhao@ucalgary.ca),

[oluwapelumi.laditan@ucalgary.ca](mailto:oluwapelumi.laditan@ucalgary.ca), [yuhua.guo@ucalgary.ca](mailto:yuhua.guo@ucalgary.ca) and [tobi.lawal1@ucalgary.ca](mailto:tobi.lawal1@ucalgary.ca)

1. **Introduction**

Nowadays, people majorly depend on automobiles for their daily commute because of its flexibility, privacy and time saving benefits, and it also provides social benefits, where it is used as a symbol of high standard of living [1]. This invariably has led to high volume of traffic in urban areas and highways. In turn, there would be a growing increase in road accidents caused by several factors and amongst this factor, a potent one to consider today is driver's drowsiness. Drowsiness is identified as one of the major causes of fatal traffic accidents. Unfortunately, about 20% of drivers tend to show drowsiness while driving, reported by National Safety Council[2]. Also, the American National Highway Traffic Safety Administration reported that drowsy driving is related to at least 100,000 motor-vehicle crashes with more than 1,500 deaths per year and the estimated annual monetary loss related is estimated to be about about $12.5 billion[2]. This statistic is so alarming hence the need to use deep learning to provide a permanent solution.

Amongst several solutions proposed from research, for example using sensor applications like EEG and detecting physiological features from drivers[13], one very simple and effective way to reduce the number of accidents is early detection of driver drowsiness and alerting with an alarm.

This project aims to apply this solution by combining a fine-tuned neural network and python-based face detection and feature extraction module into a real-time drowsiness detection system that will improve road safety.

Current state-of-the-art facial expression recognition models are able to achieve an accuracy of around 75-80%, utilizing the VGG-16 model [11]. In this project, we will be using the mobilenet model architecture to better train our model to improve the accuracy of facial and feature recognition. Considering that drowsiness detection is arguably easier to differentiate, we would consider an acceptable model performing with 70% validation accuracy and accuracy greater than 75% being widely successful. Also, we will provide a real time detection system working on the principles of openCV to allow for possible deployment or application in several automobile ranges.

The primary motivation for choosing this topic is to apply deep learning concepts and techniques we learned to a real-life problem with practical use.

1. **Material and Methods**

* 1. Dataset

In the last stage, we have used a [Kaggle dataset](https://www.kaggle.com/serenaraju/yawn-eye-dataset-new) containing 2900 images with four labels: closed, open, no\_yawn, and yawn. Those four labels represent human face images with eyes closed, eyes open, without yawn, and with yawn, respectively. The four classes in the dataset are balanced.

As described in the midterm report, we have developed several models with the best validation accuracy of 0.7098. As initially planned, we worked on finding a new dataset with more data to resolve the overfitting issue presented in various models with deep architectures.

In this stage, we have found another dataset containing 4000 images covering two classes, eyes closed and eyes open and the dataset was splitted into 3600 train sets and 400 validation for model training. However, we are not able to find the same quantity of images with classes yawn and non-yawn. Due to limited time constraints, we decided not to make or search for 4000 yawn and non-yawn human face images. However, as stated in section 3, we achieve good accuracy in detecting drowsiness based on the binary classification model we built with the new two classes dataset.

2.1 Flow Chart

In the last stage of the project, we presented a simple minimalistic design approach to tackling this grievous problem and it comprised, of face detection, feature extraction and model training, however, a more robust approach has been implemented to incorporate the real time detection feature as shown in figure 1 below. A webcam was added to collect real time face data of the driver, More so, viola jones haar cascade and OpenCv was used to improve eye detection and draw the bounding boxes around the eye location.

* 1. Face detection and Eye region extraction

As pointed out by previous works[5], eye-based methods and mouth-based methods are the two main categories of drowsiness detection methods. The original intent of this work is to train a network that classifies drowsiness based on both eye region and mouth region. However, we have focused on only eye region classification in the model and eye region recognition in the real-time detection module because of time constraints and dataset limitations.

This python module uses a webcam to capture user face images in stream. After system initialization, four OpenCV algorithms are used to detect faces, eyes, and eyes with glasses.Viola Jones face detection algorithm is used to detect the face on the image and it is then passed to the viola jones eye detection algorithm for eye detection. Once the face was detected , viola jones eye detection algorithm is used to extract the eye region from the facial images and given as an output to the convolutional neural network [1]. Figure 2 below illustrates the result of viola jones eye detection techniques. A partial image containing only the eye region is extracted and processed with the same pre-processing steps as the training data. In this work, images are resized to (224, 224), and min-max scaled using (1./225). The processed image is then fed to the trained model. A numerical value of range (0, 1) is to be returned, with 0 representing closed eyes and 1 representing open eyes.

We have designed a set of unit test test cases for this real-time detection module. The test case design covers robustness testing and worst-case testing. Detailed testing results are described in section 3.

2.4 Preprocessing and Feature Extraction

We have adopted several data preprocessing techniques, including data augmentations, min-max scaling. The current setup is the best combination that will most positively affect the model accuracy. After eye extraction is performed, it is then converted into an array of feature vectors . Features vectors are needed for each eye state drowsy image. For real time detection, feature vectors collected will be compared against a database or in a Deep learning model to be used to detect the drowsy state of the driver. Usually CNNs require fixed size images as input so preprocessing is required.

* 1. Neural network architecture design

In the project proposal, we mentioned that we had reviewed some similar works[2][3] reporting good accuracy on ResNet, VGG-FaceNet[7], InceptionV3, AlexNet[6], FlowImageNet[8]. AlexNet is fine-tuned to learn features related to drowsiness. The VGG-FaceNet is trained to learn facial features related to drowsiness. It is robust to genders, ethnicity, hairstyle and various accessories adornment. FlowImageNet takes a dense optical flow image extracted from consecutive image sequences and is trained to learn behaviour features related to drowsiness, such as facial and head movements. The plan was to train multiple networks separately and ensemble good performing networks to cover all necessary features essential to detect drowsiness[4].

In the last stage, we have implemented AlexNet and VGG-16 models. The result showed possible overfitting. To have a good performing baseline model, we decided to build a less complex CNN architecture. This model has a total of 28,600 trainable parameters and is in the process of further development. In this model, we use “categorical\_crossentropy” to define the loss function and use “accuracy” as the error metric. After 96 epoch, both accuracy and loss tend to flatten. Metrics at epoch 96: test accuracy is 0.7875, test loss is 0.4176, training accuracy is 0.7931, training loss is 0.3992, validation accuracy is 0.7098, validation loss is 0.4812.

2.3.1 MobileNet

From several research and reference to our ENEL 645 course content, we decided to implement transfer learning deep learning techniques to improve our result. With the fact that we wanted to develop a light model which can be easily deployed and need less computational resources, we decided to go for MobileNet which is a class of CNN that was open-sourced by Google[12]. More so, this model was built to be used for mobile phones, hence its lightweight nature and it is also Tensors Flow’s first mobile computer vision model[12]. MobileNet operational principle revolves around depth wise separable convolutions**,** andthis allows it to significantly reduce the number of parameters when compared to the network with regular convolutions with the same depth in the nets[12]. This process resultant is a lightweight deep neural network whose architecture is in figure

Since our final goal is to have a light yet powerful model that can be used on a mobile device in order to make real-time drowsiness detection on a driver, we shifted our focus to fine-tuning a pre-trained MobileNet. Currently, this improved MobileNet architecture is showing good results on accuracy and acceptable performance on training time, as described in section 3 of this report.

The architecture is summarized in Table 1.

1. **Results and Discussion**

This section provides result, analysis, discussion and it also includes a comparison for the offline training process between the proposed model and other models. The online operating process successfully validates the drowsiness level accuracy in real-time video stream.

The improved MobileNet we have built performs well with a generally good training accuracy, validation accuracy, and training time. During training, we have used much fewer epochs (i.e. 7 epochs) than the baseline model, since more epochs tend to overfit the model easily based on our training experiments. At epoch 7, training loss is 6.4966e-04, training accuracy is 1.0000, validation loss is 0.0671, and validation accuracy is 0.9900. A detailed training history is shown in Figure 1.

In regards to real-time testing, we do not have a numerical value for accuracy since we do not have a proper way to analyze each frame of webcam images and label and compare prediction results with ground-truth label value. However, if latency is ignored, the model prediction accuracy is approximating 100%.

In this stage, latency ranges from about 0.5 to 1 second. Since this part of the project is not closely related to the core content of this course, we decided to devote more energy into optimizing accuracy.

In the project files, we have also attached a short video with one team member testing the real-time detection module. This test video has shown a typical reaction of our detection system to closed eye state and open eye state.

**3.0.1 Offline Learning Results**

The training data has 1800 opened right eye and 1800 closed right eye image samples. The validation data contains 200 opened right eye and 200 closed right eye image samples. Table 2 lists the proposed CNN and other popular models that were used for offline learning. It shows that the proposed CNN model which is mobilenet achieves the highest validation accuracy in a short training time with 10 epochs and 32 batch size.

**3.0.2 Online Learning Results and Computational resources used**

Dell laptop with a logitech 720p webcam with 16 GB memory is used to validate the accuracy of the model in a real-time situation. In addition, Python 3.6, OpenCV 4.0 [32], Keras 2.3 are used to build the software for validating the accuracy of the drowsiness detection. also to measure the accuracy of the system , we only considered it in bright lightning conditions. Also, winsound module is used to generate distinct sound alarms to warn the user to wake up when drowsiness is detected. This model was tested on 2 different users and very similar results were recorded, and the result is shown in Table 2.

**4.0 Conclusion**

We have considered different approaches concerning the development and implementation of an algorithm to detect and alarm drowsy drivers. Mobilenet Architecture gives better accuracy compared to other models and also it is light model and has easy deployment capabilities. Furthermore, it serves as a good baseline for future training of projects related to drowsiness or its related field detection.

1. **Future works**

We intend to keep working on this project after the final presentation and final reports are delivered. We feel the potential of this work to become a better performing tool with some potential to be commercialized.

In order to achieve this goal, the following additional works have to be done:

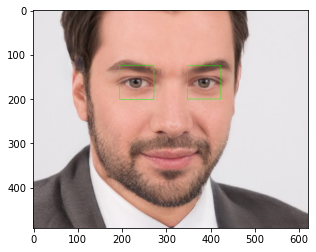
1. Add more implementations of python/OpenCV modules in order to achieve more facial feature recognition, such as motion (eye movement, face direction), mouth (yawn, talking, smoking), mood.
2. Plan to improve image cropping and to focus on the facial part only, using MTCNN(Multitask cascaded convolution neural network). MTCNN is a framework to improve face detection and face alignment and it consists of three stages of convolutional networks that can recognise faces and landmark location e.g eye, nose, left mouth corner and right mouth corner.
3. Find or make more human face images to enrich the dataset. The goal is to have a multi-class training dataset of which the class can cover all the above-mentioned facial features in separate classes, such as eyes\_not\_on\_road, not\_facing\_front, yawn, talking, phone\_call, smoking, happy, angry.
4. If the improved MobileNet does not performs well, new architectures are to be designed and implemented.
5. Testing out our model in different lightning conditions to imitate real time driving environments.

**Figures and Tables**

Figure 1. Flow Chart

****

**Figure 2. Result of the effect of viola jones techniques for face detection(extracted from testset in jupyter notebook)**

**  **

**Figure 3. MobileNet architecture**

****

**Table 1. Improved MobileNet network summary**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

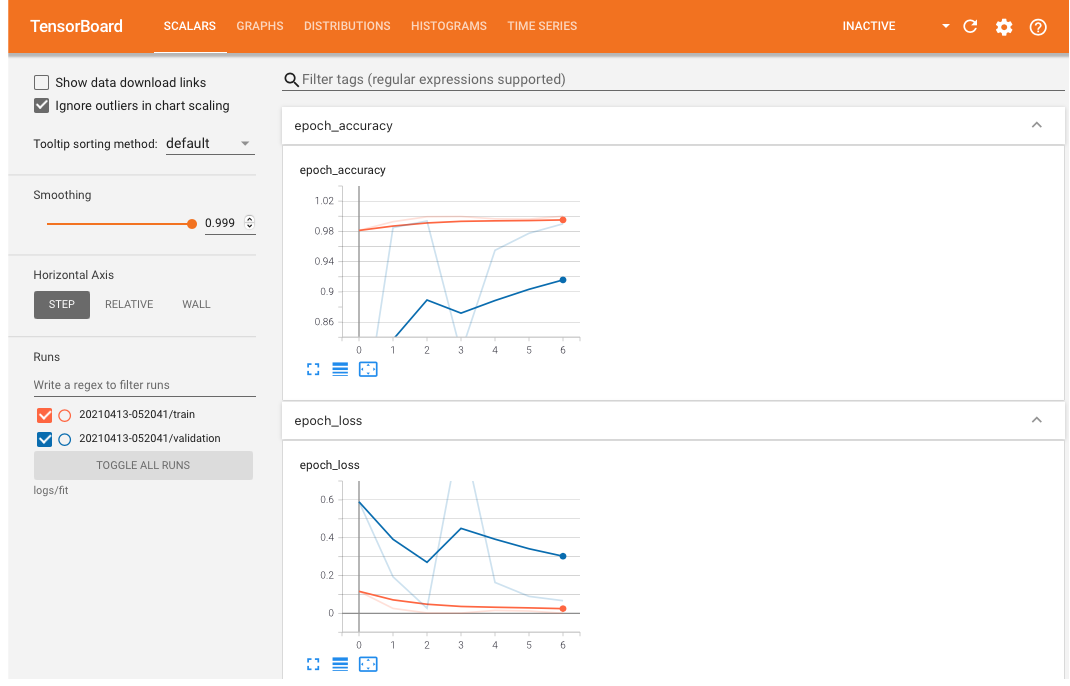
Description automatically generated**

**Table

Description automatically generated**

**Table 2. Accuracy comparison of different symptoms for different light conditions**

|  |  |  |
| --- | --- | --- |
| **Model** | **Lightning Condition** | **Validation Accuracy** |
| Mobile\_Net | Brightness | 0.9999 |
| Alex\_Net & VGG-16 | Brightness | 0.7098 |

****

**Figure 1. Tensor Board training history**

**References**

[1] Vijayan, Vineetha, and Sherly, Elizabeth. "Real Time Detection System of Driver Drowsiness Based on Representation Learning Using Deep Neural Networks." Journal of Intelligent & Fuzzy Systems 36.3 (2019): 1977-985. Web.

[2] Drivers are falling asleep behind the wheel, National Safety Council. <https://www.nsc.org/road/safety-topics/fatigued-driver>

[3] Park, Sanghyuk, Pan, Fei, Kang, Sunghun, and Yoo, Chang D. "Driver Drowsiness Detection System Based on Feature Representation Learning Using Various Deep Networks." Computer Vision – ACCV 2016 Workshops 10118 (2017): 154-64. Web.

[4] Dua, Mohit, Shakshi, Singla, Ritu, Raj, Saumya, and Jangra, Arti. "Deep CNN Models-based Ensemble Approach to Driver Drowsiness Detection." Neural Computing & Applications (2020): Neural Computing & Applications, 2020-07-20. Web.

[5] Zhao, Lei, Wang, Zengcai, Zhang, Guoxin, and Gao, Huanbing. "Driver Drowsiness Recognition via Transferred Deep 3D Convolutional Network and State Probability Vector." Multimedia Tools and Applications 79.35-36 (2020): 26683-6701. Web.

[6] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)

[7] Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC, vol. 1, p. 6 (2015)

[8] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR, pp. 2625–2634 (2015)

[9] Weng, Ching-Hua, Lai, Ying-Hsiu, and Lai, Shang-Hong. "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network." Computer Vision – ACCV 2016 Workshops 10118 (2017): 117-33. Web.

[10] Bhargava Reddy, Ye-Hoon Kim, Sojung Yun, Chanwon Seo, Junik Jang.

“Real-time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017, pp. 121-128

[11] T. Vo, G. Lee, H. Yang and S. Kim, "Pyramid With Super Resolution for In-the-Wild Facial Expression Recognition," in IEEE Access, vol. 8, pp. 131988-132001, 2020, doi: 10.1109/ACCESS.2020.3010018.

[12]Pujara, A., 2020. *Image Classification With MobileNet*. [online] Medium. Available at: <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470> [Accessed 14 April 2021].

[3] A. Chowdhury, R. Shankaran, M. Kavakli and M. M. Haque, "Sensor Applications and Physiological Features in Drivers’ Drowsiness Detection: A Review," in IEEE Sensors Journal, vol. 18, no. 8, pp. 3055-3067, 15 April15, 2018, doi: 10.1109/JSEN.2018.2807245.

**Github Repository**

<https://github.com/ENEL645-Group6/Project.git>

**Member Contributions**

Each member had a different task and completed various sections of this work, and the workloads are distributed equally.

|  |  |  |
| --- | --- | --- |
| Team Members | Task | Status |
| Tianhan Jiang | Model Testing and Report writing | Completed |
| Peiyun Zhao | Build Model Architecture | Completed |
| David Laditan | Preprocessing | Completed |
| David Guo | Data Augmentation | Completed |
| Tobi Lawal | Data Collection | Completed |