**Group 6 Project Midterm Report**

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

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

**Contact Email**: [yuhua.guo@ucalgary.ca](mailto:yuhua.guo@ucalgary.ca), [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) and [tobi.lawal1@ucalgary.ca](mailto:tobi.lawal1@ucalgary.ca)

1. **Motivation**

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

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[1]. This project aims to combine a fine-tuned neural network and a python-based face detection and feature extraction module into a real-time drowsiness detection system that will contribute to improving road safety.

1. **Methodology**
   1. Dataset and data preprocessing

In this stage, we use a [kaggle dataset](https://www.kaggle.com/serenaraju/yawn-eye-dataset-new) that contains 2900 images with four labels, which are 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.

In next stage, we may add more image data to the dataset to improve accuracy. If time permits, we might also increase other labels to further categorize the spectrum between drowsy and non-drowsy.

* 1. Neural network architecture design

In the project proposal, we mentioned that we have 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, which 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 this stage, we have implement AlexNet. The result shows a possible overfitting and training time complexity performance is not well. Since other networks are likely to have even deeper architectures than AlexNet, they are likely to return an overfitting result as well.

We decided to build a less complex architecture until more suitable data is found and added on top of our current dataset. This network will also serve as the baseline for all other networks we are going to explore in the next stage. A summary of the baseline network is shown in Table 1.

Detailed description of hyperparameter tuning and the metrics are covered in section 3.

* 1. Face detection and feature extraction module

As pointed out by previous works[5], eye-based methods and mouth-based methods are the two main categories of drowsiness detection methods. We plan to cover both aspects by using multiple networks.

We plan to implement a python-based module to preprocess all image from training set, validation set, and test set. This module will be used to perform a facial recognition first, keep only facial region of the image.

This module will also contain algorithms to extract eye regions and mouth regions separately. For training set and validation set, either eye feature extraction or mouth region extraction functionality will be used to keep only the region that matches the label. For test set, or real-time drowsiness detection, we will use both functionalities to extract both eye and mouth region features and perform classification.

A scheme of this process can be find in Figure 1.

1. **Preliminary results**

Currently, the network with best performance in accuracy and time complexity is the one with architecture descried in Table 1 (denoted as baseline model).

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 flattening. 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.

1. **Progress and future plans**
   1. Current progress

As stated in section 2.2, we have tried several deep architectures that all ended up giving overfitting results. Models with greater number of layer, the ones with multiple dense connected layers, or the ones with too many parameters will take a relatively long time to train (from 3 minutes to 35 minutes per epoch).

After manipulating with parameters and models, we had the following findings:

Parameters that will not improve time complexity:

* batch\_size parameter of ImageDataGenerator,
* batch\_size parameter of fit function,
* color\_mode (rgb or greyscale),
* with or without data augmentation,
* learning rate,
* training steps,
* validation steps.

Parameters that are relevant to time complexity:

* number of convolutional layers
* number of filters in each convolutional layer
* number of fully connected layers
* output parameters of dense layers
* if number of parameters of any layer is too many
  1. Other model architectures to be implemented
  2. Python feature extraction module

**Figures and Tables**

**Table 1. Baseline network summary**

**Table

Description automatically generated**

**Table 2. Project progress by midterm**

|  |  |
| --- | --- |
| **Task** | **Status** |
| Data Collection | Completed |
| Preprocessing and Data Augmentation | Completed |
| Build Model Architectures | Completed, more fine-tuning needed |
| Model Testing | Completed on existing networks |
| Final Report and Video Recording | Not started |

Video Input/Camera/Database

Face Detection

Eye/Mouth Region Extraction

Feature Extraction

CNN Classifier

Drowsy

Non -Drowsy

**Figure 1. Our Proposed System Architecture**

**References**

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

[2] 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.

[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

**Member Contributions**

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