Driver Fatigue Detection for Autonomous Vehicle Paradigm

Zhaoyu Yin

20010277

Project Type: Thesis

Course Code: CISC 499

Supervisor(s):

Farhana Zulkernine

Date: April 10, 2020

Abstract

Detecting the driver's status is important because one of the main reasons for motor vehicular accidents is related to driver's fatigue or inattention. Many studies have been conducted to reduce the accidents since the last decade in various areas. However, some of them could be limited by the past technologies. With the development of artificial intelligence and visual sensors, numerous accidents can be efficiently avoided. The real-time fatigue detection is necessary as a moment of negligence may happen at any time. The detector could be easily accessible and deployable. Any device with sufficient computing power and a high definition camera is able to implement the real-time fatigue detection. In this report, an approach towards real-time fatigue detection based on deep learning which can be implemented on a laptop and performs with a high accuracy is proposed. To implement the detection, the data set was preprocessed and divided by each label. Moreover, a neural network was designed based on facial pattern input to detect whether a driver is drowsy or not. The proposed model achieved an accuracy of 83% on the yawning classification.

Table of Content

Introduction

1.1. Motivation

According to the National Highway Traffic Safety Administration, every year about 100,000 police-reported crashes involve drowsy driving, resulting in injury or death that costs society $109 billion annually. Even in Japan, attention lapse, including that due to driving while drowsy, was the primary reason for traffic accidents in many years. No doubt proper techniques need to be applied to reduce the lost, both in economy and human life. One solution to handle the problem is the development of an intelligent device that can predict driver drowsiness and avoid drowsy driving. The initial idea was proposed many years ago and the scientists conducted researches in many directions. However, with the technology restrictions, only a little progress was made until the recent decade. Nowadays, both the miniaturization of cameras and the evolution of video resolution make real-time detection step into reality. My motivation is trying to process and improve current methods of data analytics, making it possible to detect no matter whether the driver is facing the camera in right angle or not. Real-time processing is another challenge as the vehicular hardware is required to handle video data processing and send warnings at the same time. Only important features will be retained to simplify the calculation. As a result, the system can relieve drivers from struggling against drowsiness by detecting their states and keep them awake. Besides, extracting certain patterns from the image can be applied to other problems such as emotion recognition, identity recognition, and so on. In the future, I believe the vehicle can understand our feelings from the emotions and can even be unlocked once it scans our faces.

1.2. Problem Description

To implement driver drowsiness detection, three challenges need to be overcome. The first challenge is how to preprocess the data in YawDD properly. Considering that there is no labelled dataset we can use directly, it is necessary to extract images from videos and label them manually. The second problem I need to explore is figuring out a method to build a model that can detect fatigue efficiently and accurately. Last but not least, detecting static images with high accuracy is not sufficient. To realize the real-time fatigue detection, the system should be able to process instant video signal input from the camera. Additionally, the fatigue detection looks similar to a small application. The content will be displayed clearly on the desktop as a window. At the same time, although the detection is processing instantly and indicating many feature points on the face, the system must assure operating smoothly. In another word, the detection needs to be practical and user-friendly.

1.3. Key contributions

In summary, I researched many papers and found a practical method to realize the function of driver fatigue detection. First of all, I succeeded in preprocessing the dataset, slicing and labelling images extracted from videos. Then, I determined to use the ResNet50 structure and train this model with the preprocessed data. Additionally, what really counts is to achieve a balance between the accuracy and computational power in the progress of real-time fatigue detection. Considering that the camera can hardly be installed directly towards the driver's face under real circumstances, it is highly meaningful to detect the driver's state from different angles.

1.4. Organization

In the following part, I will introduce the related work such as the development of the fatigue detection and the outcome of my literature studies. Then, the detailed implementation which includes data set, model selection, detection realization will be shown and analyzed respectively. Results and related tests will also be displayed. I will fully explain the reason of each procedure and why I make such choice. Last but not least, as there are several restrictions in this project, the limitations, the future work will be discussed and displayed after the conclusion in the final part. At the end of the report, all related papers and techniques I used will be listed in detail.

Background

2.1 Background

Nowadays, as vehicle safety technologies are developing at high speed, automobile manufacturers concentrate more on not only preventing external impact but also detecting driver drowsiness. Various techniques have been implemented to measure driver drowsiness, which can be broadly classified into three categories such as Driving pattern of the vehicle, Psychophysiological characteristics analysis, Computer Vision techniques for driver monitoring. In the first group, various advanced techniques are applied to monitor steering wheel movement such as the acceleration. The second category focuses on electrical bio-signals such as EGG, ECG and so on.

However, the techniques in the two previously mentioned classes have severe limitations. The former class of techniques can only be used in certain condition, whereas the latter one is uncomfortable and inconvenient for drivers. Thus, driver monitoring based on computer vision is becoming popular not only in consideration of cost but also low impact on drivers. Even using a single camera can meet the requirement for minimum configuration. The detection does not influence driving habit as well. Last but not least, compared to other immature techniques, using cameras to process image data instantaneously can achieve better accuracy and send a warning about drowsiness to driver with low delay. Thus, it is possible for the driver to avoid the accident.

When it comes to computer vision techniques, there are several aspects to realize fatigue driving detection as well. The simplest method is to calculate eye's blink in a period of time. Regardless, this method has many limitations. Different people can hardly have the same frequency of eyes blinking. If the frequency of eyes blinking exceeds a threshold, the driving will be determined as fatigue driving. Another aspect is to track eyes' or mouth's movement. It has become the main trend recent years. Next comes my literature studies in relevant computer vision aspects.

2.2 Literature Study

1. Real-time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks [1]

To improve the speed of drowsiness detection and accuracy, the author followed three procedures. Convolutional approaches on drowsiness detection are listed, followed by the latest approaches using the deep learning. Then, the author described the compression algorithms to overcome run-time issues.

Method

The overall architecture of the proposed drowsiness detection consists of two steps. The first step is the joint face detection and alignment. The second is the drowsiness detection model. Multi-Task Cascaded Convolutional Networks (MTCNN) [2] is used as it is one of the fastest and accurate face detectors. It can achieve high speed in joint face detection. Driver Drowsiness Detection Network (DDDN) is used for detecting driver's drowsiness. DDDN takes the output of the first step (face detection and alignment) as its input. Next are the three models authors used.

- **Baseline-4 Model:** A 4-stream deep neural network. The inputs consist of left-eye, right-eye, mouth and face.
- **Baseline-2 Model:** To improve the speed, it only contains the left-eye and mouth.
- **Compressed-2 Model:** To improve the speed further, the technique introduces two concepts, namely teacher network and student network.

Result

Baseline-2 model achieves the best at 93.8%.

The compressed-2 model achieved around 90% by using 3 times lesser disk space than baseline-2 model.

2. Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques [3]

The authors used the Driver Drowsiness Detection Dataset from NTHU. For data processing, Multilayers Perceptron Classifier which can also be referred as MLP is used to help secure the minimum of errors possible until the needed input-output mapping is achieved.

Method

Fig.2.2.1 Procedures of detecting drowsiness

- **Extracting Videos from NTHU Database**: 18 subjects and 4 subjects are used in the Training dataset and the Evaluation Dataset.
- **Extracting Images from Video Frames**: The rate of selecting videos from the dataset is 30 frames per second; the author extracts every video frame as images.
- **Extracting landmark coordinates from images:** Dlib is used to extract landmark coordinates from images.
- **Training the Algorithm:** The landmark coordinates extracted from images will act as the input to the algorithm, based on Multiply Perceptron Classifier with three hidden layers.
- **Model Extraction:** Finally, the algorithm can decide if the driver is drowsy or not based on his or her face landmark.

Result

Regarding accuracy, its value when physiological sensors are used ranges from 73% to 86%. 81% represents one of the best results regarding efficiency.

3. Vision-based drowsiness detection for Real Driving Conditions [4]

This paper presents a non-intrusive approach for monitoring driver drowsiness, based on computer vision techniques, installed on a real car. As to the data set, researchers designed an on-board system to collect the database and chose a car equipped with an IR illumination system for night conditions.

Method

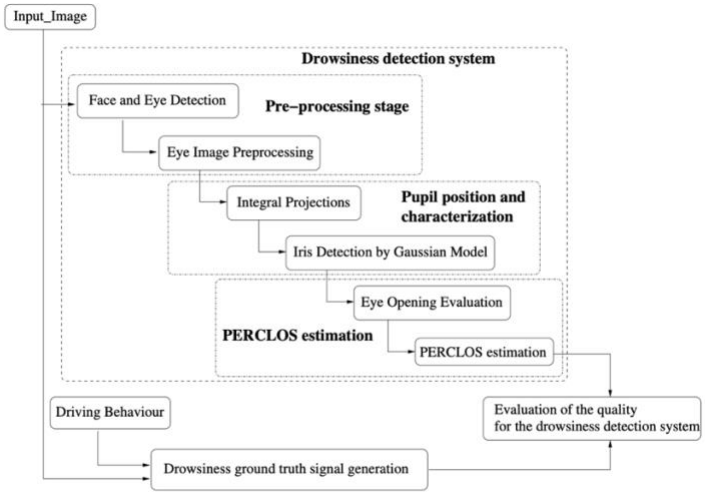


Fig.2.2.2 Structure of Drowsiness detection system

The input image is processed to detect face and eyes. The Kalman filter is used to track their position. Once the eyes are detected, iris centre location module is executed. It is based on integral projections. Then the eyes closure is calculated by using a Gaussian Model.

1. Face and Eye detection: The eyes position will be located throughout the whole image sequence. Eye searching is carried out in two steps. The first one consists in looking for the face. Once the area is found, the eye detector is applied on it.

2. Eye Image Filtering: Once the ROI is calculated for each eye, morphologic transformations are applied to remove bright artifacts. Therefore, a normalization to enhance gray levels is used.
3. Eye Closure Evaluation: The eye closure evaluation algorithm is composed of integral projections, horizontal and vertical, from the detected eye area, to have an estimation of the iris centre.

Result

The PERCLOS [5] estimation method has demonstrated to be accurate, showing specificity of 92.21% and a sensitivity of 79.84% in average.

4. Analysis of Real Time Driver Fatigue Detection Based on Eyes and Yawning [6]

The authors proposed the work by applying the behavioral or visual based measures. The driver's behaviours such as eye blink, head movement, and yawning are monitored to examine the state. The system will alert the driver with a buzzer once the frequency of closed eyes or the open mouth simultaneously exceeds the threshold.

Method

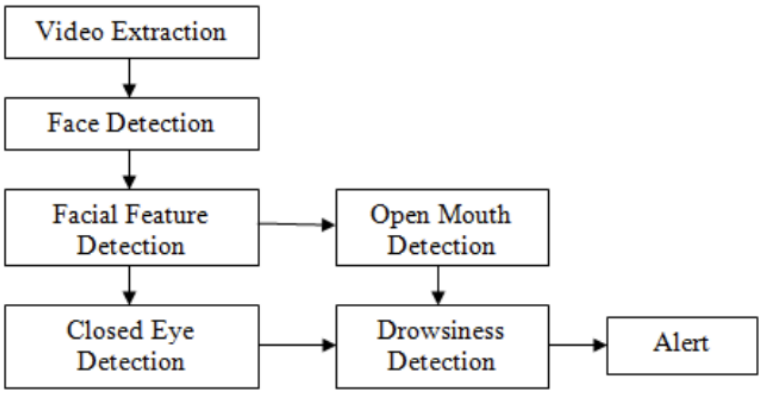


Fig.2.2.3 Procedures of detecting fatigue based on eyes and yawning

This study gives a blend of yawning and eye blink detection altogether. Next follows each step to implement the detection.

1. Face and eye detection: In this phase, the method 'Viola Jones' [7] is applied to detect the face and eyes, with a training set of faces and eyes provided in OpenCV. After getting the center of the eye, we define the eye ROI in the following way.
2. Eye blinking detection(open/close):
    - Detect eye state
    - Detect the eye blinking rate
3. Mouth detection(open):
    - Mouth Detection
    - Segmentation and Smoothing
    - Detect the contour of the mouth
    - Decision of yawning

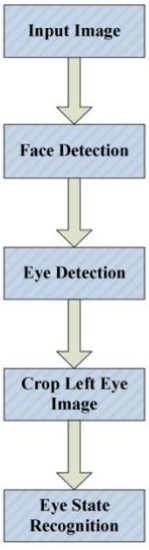
Result

The positive alert was best recorded in critical time 1 (88%) for 50-60 age driver and the error was more for the with moustache in critical time 2(29%) for 18-25 age driver.

5. An Eye State Recognition Method for Drowsiness Detection [8]

The researchers present a method for recognizing eye state using eye blinking information. Lots of approaches have been proposed in the field of face detection. Then features will be extracted from the input image. The boundary of eyes can be calculated based on their coordinates. Then the author gets the left eye's features using the Local Binary Pattern method and adopt the SVM classifier to determine the eye state.

Method



A. The Face Recognition

B. The Detection of Eyes Locations

C. The LBP Feature for Left Eye: Real drowsiness is un-intentional as the left eye and the right eye keeps same states no matter they are open or close, so it is reasonable to consider only one eye's state.

D. The SVM Classifier for Eye State Recognition: The SVM is used for two-class classification.

Result

The accuracy of closed eye and open eye are respectively 87.13% and 93.62%.

6. **Driver Drowsiness Recognition Based on Computer Vision Technology [9]**

The paper presents a nonintrusive drowsiness detection method using eye-tracking and image processing. A robust eye detection algorithm is introduced to address the problems caused by changes in illumination and driver posture.

Method

1. Eye movement detection

When the driver is facing the camera directly, an AdaBoost-based face detector can be applied to determine the position and boundary of the face. An Active Shape Model (ASM) [10] algorithm is used to precisely locate the human eyes from the front-view images.

2. Drowsiness Features

It is shown that the eye movements differ dramatically for various drowsiness levels. The fully awake state is characterized by low frequency, high speed movements with larger mean amplitude.

3. High Fidelity Driving Simulator

The visual display technology and high-fidelity audio system enhance the driving experience. The test subject is immersed in sight, sound, and movement environments so real that impending crash scenarios can be convincingly presented with no danger to the subject.

Result

Average 80% for Drowsy, 78% for very Drowsy

7. A New System for Driver Drowsiness and Distraction Detection [11]

In order to achieve better accuracy in face tracking, the paper proposes a new method which is combination of detection and object tracking. As to the face tracking, it has capability to self-correction. After eye region is found, Local Binary Pattern (LBP) [12] is employed to extract eye characteristics. Using these features, an SVM classifier was trained to perform eye state analysis.

Method

A. Face Detection: Proposed system uses Viola and Jones (VJ) method which based on machine learning approach for visual object detection. This method takes advantage of three different features that are integral image, AdaBoost technique and the cascade classifier.

B. Object Detection: We apply Ning et al. method to track the object. This method is based on mean shift. In addition to use color histogram, this method utilized LBP pattern and propose joint color-texture histogram method which represent target very distinctive and effective.

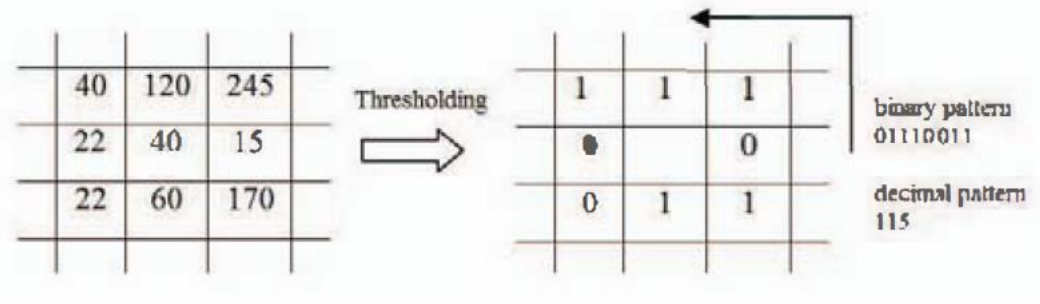


Fig.2.2.4 LBP to determine the threshold

C. Feedback System: As realized the object tracking in proposed face tracking subsystem could diverge or loose target, to overcome this problem, we proposed a feedback system to detect any divergence from true position at the output and correct this error.

D. Decision Rules

Eye State Analysis: According to the efficiency and low computational time of Support Vector Machine (SVM), proposed system uses this method to analyze eye state. After eye has been detected, LBP operator will be used to extract eye characteristics.

Result

Track face by an accuracy of 100% and detecting eye blink by accuracy of 98.4%

8. Human and Vehicle Driver Drowsiness Detected by Facial Expression [13]

The researchers determined 17 features points on the face for detecting driver drowsiness, according to the facial muscles. Then, they discovered that eyebrows rise caused by contraction of frontalis muscle was important feature to detect drowsiness during driving. The Active Appearance Model was applied to measure three-dimension coordinates of the feature points on the facial image. Further, K-nearest-neighbour was applied to classify 6 drowsiness levels.

Method

EARLY-STAGE DROWSINESS DETECTION

DROWSINESS DETECTION USING FACIAL EXPRESSION

A. Features of drowsy Expression: We divided the reference states of drowsiness into 6 levels, i.e., “Not Sleepy”, “Slightly Sleepy”, “Sleepy”, “Rather Sleepy”, “Very Sleepy”, and “Sleeping” by adding the “Sleeping” level to Kitajima’s trained observer rating scale since we found some participants dozed off during simulated driving at the preliminary experiments.

B. Measuring the characteristics of drowsy expressions from facial images: This method is based on an active appearance model (AAM) [14], which detects the 3-dimensional coordinates of the measurement points of the driver's face every frame. Our AAM consists of a specific 2-dimensional model and a general 3-dimensional model.

C. Method to detect sleepiness: We use the k-nearest neighbor method (which is one of the pattern classification methods) to detect sleepiness.

DETECTING AWAKE EXPRESSIONS

Result

As it is estimated, the results of the drowsiness detection correspond to the drowsiness reference by a trained observer with an average RMSE of less than 1.0 level among 13 participants.

9. Machine Learning Systems for Detecting Driving Drowsiness [15]

The researchers employ machine learning to datamine actual human behaviour during drowsiness episodes. They develop an automatic classifier which can classify 30 facial actions from the facial action coding system. These facial actions consist of blinking and yawning motions. Thus, the system can detect driving drowsiness.

Method

Head movement was measured using an accelerometer that has 3 degrees of freedom.

The facial motion coding system is arguably the most widely used method for coding facial expressions in behavioral science. The system describes facial expressions based on 46 component movements, which roughly correspond to individual facial muscle movements. FACS provides an objective and comprehensive method to analyze expressions as basic components, similar to decomposing speech into phonemes.

Facial Action Signals: The output of the facial motion detector consists of a continuous value for each frame, which is the distance to the separating hyperplane.

Drowsiness Prediction: The facial motion output is passed to the classifier for predicting drowsiness based on the automatically detected facial behavior. Two learning-based classifiers Adaboost and polynomial ridge regression are compared. Both tested intra-subject predictions of drowsiness and cross-subject predictions (subject-independent) of drowsiness.

Result

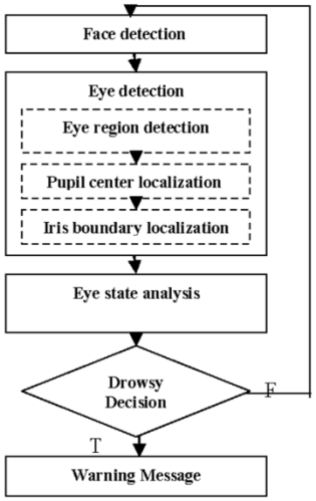
The system was able to predict sleep and crash episodes on a simulator with 98% accuracy across subjects.

10. Open/Closed Eye Analysis for Drowsiness Detection [16]

The researchers propose a method for eye detection by using eye map, thus achieving excellent pupil center and iris boundary localization results on the IMM database. They also applied an

eye state analysis algorithm using five video sequences and show superior results compared to the common techniques.

Method



1. Face Detection Algorithm

2. Eye Detection Algorithm

   Eye Region Detection Algorithm

   Pupil Center Localization Algorithm

   Iris Circle Detection Algorithm

3. Eye State Analysis

   Eyelids distance-based technique

   Introducing Chromatic-based algorithm

Drowsy Decision

The decision boundary for the drowsiness is empirically 4000 obtained by graphs from 5 persons in non-drowsy and drowsy state. When the parameter exceeded 40°O, person is given a warning message for the drowsy state.

Result

The average accuracy of detecting open and closed eyes respectively reach 98.8% and 96.4%.

Implementation

3.1 Overview

To solve the problem, the approach seems clear enough and can be divided into several main procedures. Briefly speaking, I will preprocess the data in YawDD [17] at first and train a model with the preprocessed data for implementation. The dataset consists of many videos so that I have to extract images from these videos. To detect the driver's face in each image, Dlib [18] will be applied to detect the face and resize the image.

After preprocessing the data, a neural network will be built. Considering that the network must be able to process massive image data, I prefer to choose convolutional neural network as it uses special structures for image recognition. Because of its high speed, it is easy to use multi-layer neural networks. Meanwhile, the multi-layer structure has a great advantage in recognition accuracy. Among convolutional neural networks, the ResNet [19] is considered as the most suitable structure in this approach. ResNet (Resident Neural Network) was proposed by KaiMing He and other four Chinese researchers from Microsoft Research Institute. The 152-layer neural network was successfully trained by using ResNet Unit. Compared to VGGNet, it has a smaller number of parameters. The main idea of ResNet is to increase the direct connection channel in the network, which is similar to Highway Network. The previous network structure was a non-linear transformation of the performance input, while Highway Network allows to retain a certain proportion of the output of the previous network layer. The idea of ResNet is very similar to that of Highway Network, allowing the original input information to be passed directly to the subsequent layers, as shown in the following figure.

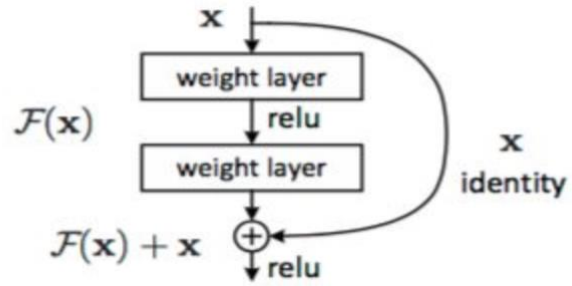


Fig.3.1.1 Residual Learning Unit

Compared to VGG19 network, the residual unit is added through the short circuit mechanism. The change is mainly reflected in ResNet directly using stride = 2 convolution for down sampling and replacing the fully connected layer with the global average pool layer. An important design principle of ResNet is that when the size of the feature map is reduced by half, the number of feature maps doubles, which maintains the complexity of the network layer. Compared with the ordinary network, ResNet adds a short-circuit mechanism between every two layers, which forms residual learning.

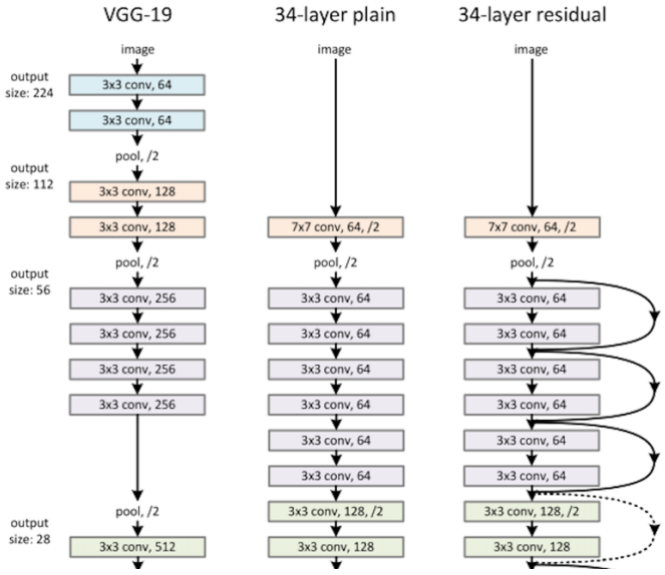


Fig.3.1.2 Parts of VGG and ResNet34 Structure

The traditional convolutional network or fully connected network is likely to suffer from information loss along with gradient diminishing or gradient explosion, resulting in the failure to train a deep network. To some extent ResNet solves this problem. By directly passing the input information to the output, the integrity of the information is protected. The entire network only needs to learn the difference between the input and the output, which simplifies the learning process. In this implementation, I will use a residual network with 50 layers.

To validate the work, the dataset will be divided into two parts, training set and test set by the ratio of 80%-20%. In this experiment, the accuracy and the precision of each label will be calculated. In order to apply the model for detecting fatigue instantly, I took a method similar to PERCLOS. The PERCLOS is aimed at calculating the times of eyes' blinking in unit time. The OpenCV was applied here to capture the user's face from streams, marking a square boundary around the face. Meanwhile, the model was applied to evaluate the user's state. By calculating the number of the frames that is determined as yawning in unit time, the system can make preliminary results.

3.2 Data

After researching many accessible datasets on the Internet, I think that YawDD meets my requirements perfectly. This dataset can be downloaded from ACM Mutlimedia Systems Conference Dataset Achieve. YawDD contains two video datasets of drivers with various facial characteristics, to be used for testing algorithms and models for mainly yawning detection, but also recognition and tracking of face and mouth. In the first dataset a camera is installed under the front mirror of a car. There are 47 male participants and 43 female participants in this dataset. Each participant has three or four videos and each video contains different driving conditions such as normal, talking and yawning. The total dataset size is over 5 Gigabytes. All videos are recorded in 640x480 24-bit true color (RGB) 30 frames per second AVI format.

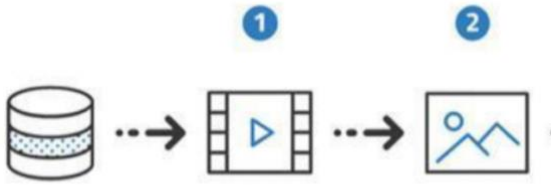


Fig.3.2.1 Extracting images from Dataset

The first procedure is to extract images from each video. The initial videos are in 30 fps AVI format. After slicing the videos, only one image with PNG format will be extracted every 10 frames, which means three images are generated every second. Then each video file is converted to a folder labelled with the participant's name and the condition, consisting of hundreds of images. The second step is to reorganize the images in these folders. Irrelevant attributes such as sunglasses, hat and so on will be ignored. All images are assigned to new

folders labelled with either normal, talking or yawning. The third procedure is to divide three types of images by the ratio of 80%-20% as training and testing set.

Fig.3.2.2 Face Capturing and Image Resizing

However, as the driver's face only occupies one fifths of each image, using the whole image is overabundant. Also, if we don't resize the image, the dataset size will exceed 6 Gigabytes, which severely increases unnecessary calculation expenses. In order to extract drivers' faces and resize images, Dlib and OpenCV [20] are applied in this procedure. Compared to OpenCV, Dlib is much more powerful to recognize the driver's profile face. Once detected, only face area will be saved thus the initial image is resized to 200x200 resolution. Although most of images can be captured by Dlib, there are still a small proportion of images that cannot be recognized. Here is the table that calculates the number of images captured successfully.

Dataset	Label	Nbr. Images Extracted	Nbr. Images not detected by Dlib
Training	Normal	4037	1048
	Talking	5892	1443
	Yawning	1009	221
Testing	Normal	995	223
	Talking	1475	340
	Yawning	250	74
Total		13658	3349

Fig.3.2.3 Numbers of images Extracted

According to the similar application in other papers, the success rate of detecting drivers' faces behaves as expected. Over three fourths of images are successfully detected and resized, providing enough samples for the following training.

3.3 Implementation Details

3.3.1 Implementation Environment

In this experiment, the laptop I use is equipped with CPU i7-6700hq, GPU GTX1070 with 8 Gigabytes video memory and 16 Gigabytes memory. GTX 1070 was used for training and testing. It has 1920 CUDA cores, each core with a base clock speed of 1506 MHz; it can do computations up to 6.5T. The operating system runs on the laptop is Windows10. When it comes to the software and libraries, Pytorch with version 1.4.0 and Torchvision with version 0.5.0 are installed. In consideration of excellent practicality and ease to use, Pytorch is the best choice to implement the network. Also, it is fully readable and suitable for processing. RGB image Besides, the system is equipped with the latest OpenCV, Dlib, Pandas, Numpy and other relevant libraries. In order to maximize the GPU performance, CUDA version 10.1 is installed. Starting learning rate of 0.001 is taken since the initial weights are all randomly initialized, taking 1000 epochs totally. When the training occurs, over 2 Gigabytes memory was used. The whole process of training the model took over 10 hours.

3.3.2 Validation

As is described above, to train the model and validate its efficiency, the dataset was divided into two parts by the ratio of 80% to 20%. There are about 10000 images in training set and over 2000 images in testing set. The architecture of my model consists of 5 stages. The first stage contains normal convolutional layers, batch norm, ReLU layers and maxpooling layers. Next follow 4 similar stages integrated with a convolutional block and different number of cars. In the end, there are avgpool layers, flatten and fully connected layers before the final output. In order to validate if the model can perform as expected, I design several experiments.

- Experiment 1:

  After debugging and making sure that all functions can work successfully with a small proportion of testing samples, I tried to change the parameters that may have an effect on the outcome. First I set epoch as 10 and learning rate as 0.1. The entire process took not so much time, but it resulted in a very low accuracy and a large loss. Next I modified the epoch and learning rate to 100 and 0.01, the results dramatically changed. The loss value declined from 0.67 to only 0.17 as the epoch increased. However, I found the total accuracy was still low, around 60%, which did not make any sense. Because the dataset was divided into three labels, normal, speaking and yawning. Among them, the speaking samples occupy the largest proportion while the yawning samples only occupy 10% of the total. Then I calculated the recall value respectively and achieved a much better accuracy of yawning detection. Thus, I could conclude that the model is indeed useful for the objective.

- Experiment 2:

  To test if the model can succeed in capturing one face and detecting the state of the person in the image. I downloaded several pictures that were not in the dataset from the Internet and used the model to evaluate them, which proved the system did not have limitations of its functionality.

3.3.3 Results

As is described in data section above, there are 2720 images detected by Dlib successfully in the testing set totally. Each label representing different conditions occupies 995, 1475 and 250 respectively. Among them, speaking samples took the largest proportion. Below displays a matrix table that calculates the number of each label.

		Actual		
		Normal	Speaking	Yawning
Predicted	Normal	646	606	12
	Speaking	306	745	31
	Yawning	43	124	207

Fig.3.3.1 Predicted and Actual number of each label

Total Accuracy: (646 + 725 + 207) / 2720 = 0.580147%

Precision of Yawning: 207 / (43 + 124 + 207) = 0.55348%

Recall of Yawing: 207 / 250 = 0.828%

Specificity of Yawning: 2152 / 2470 = 0.932389%

Obviously, we can easily reach the conclusion that the model can hardly detect the difference between normal and speaking. With strong features in yawning images, the model was trained efficiently for this label. The recall of yawning is not so bad as 207 yawning images were detected correctly. However, as the number of speaking images is so large, almost occupying half of the testing set; it is a pity that the incorrect detection of speaking lowered the total accuracy. Surprisingly, the specificity value exceeds the recall value. In another word, compared to detecting yawning condition, the mode seems better at detecting images that are not yawning. The detailed reasons why I got such results will be discussed in the following sections.

3.3.4 Discussion

In this implementation, I did meet several challenges and some of them hindered me to achieve a better outcome. When preprocessing the dataset, I suddenly found that not every segment in the video is corresponding to its label. For example, the driver may not speak in a video labelled with ‘Speaking’ for half the time. He could just finish talking or be ready to talk. In another word, slicing the video labelled with ‘Speaking’ can result in generating many images that should have been labelled with ‘Normal’. The same issue exists in yawning videos as well. Luckily, I cut out the irrelevant part in yawing videos manually, trying to minimize the negative effects. Furthermore, as speaking is a complete progress, it is inappropriate to train a model by using a method for analyzing static images. Both of the factors above resulted in low accuracy of speaking detection unsurprisingly. Although detecting speaking is not the purpose of my work, it might still influence the

outcome of fatigue detection to some extent. To solve the problem, I can cut off unnecessary parts in each video and only keep important segments. Also, when preprocessing the dataset, I just used the ready-made network to capture faces. I should have trained the network with small proportion of samples and then capture all images. Another problem occurred when training the network. At first I tried to extract features such as eyes and mouth respectively instead of training the model with whole faces. However, maybe the reason for not facing the camera directly, the eyes parts can hardly be extracted. Each of these steps introduces possible loss in accuracy and precision. To solve the problem, another network should be applied and trained to detect drivers' eyes. The last challenge occurs in real-time fatigue detection. As the model is trained for analyzing static images, it is difficult to determine the threshold of fatigue in a specific process. Also, with low GPU utilization on the laptop, the stream looks very incoherent with low frames. The reason might be the poor compatibility between the detection by OpenCV and the model.

Conclusion and Future Work

4.1 Summary

The first chapter introduces the reasons why I am interested in such project and what inspires me to conduct researches in the related filed. Also, this chapter explains the objective of this project; what I can implement after I complete this project and what I can contribute to the human community. Last, there is a guideline in the end to explain what will be displayed in the next few chapters.

The Background section introduces the current three aspects for achieving fatigue driving detection and the reasons why I chose visual data analytics as my aspect instead of other two aspects. It also contains an important part Literature Review which let me understand basic methods and useful points.

The implementation part introduces the entire process of reaching the objective, including preprocessing dataset, training the network, validation and results. To conclude, the system behaves badly at distinguishing speaking and normal but good at distinguishing yawing and other types. Besides, it fully explains the reasons why I choose to use them. Last but not least, in discussion part several challenges I met in the implementation and the potential improvement are listed one by one.

The last chapter gives a summary and introduces many limitations in this project; what I have not done yet; what I will do in the future to improve the prototype and my imagination about the development of current technologies in the future.

4.2 Limitations

There are several limitations in the project. Many participants in the dataset wear a pair of sunglasses. Most of them have various shapes, which increases the difficulty of capturing drivers' faces. Also, the video can only record the driver's profile face because the camera is installed under the front mirror of the vehicle. As we can see from the above, over 20 percent of the images cannot be detected and resized by Dlib due to these factors. On the other side, due to the different weather conditions in the dataset, the differences in drivers' skin colour become much larger, which exacerbates the failure during detecting. As to the dark weather, the face landmark selector has the probability to outline drivers' faces incorrectly. When it comes to the instant detection, the model tends to slice the real-time video and evaluate the state of each frame. If the frequency of frames identified as yawning exceeds some point, the system can say that the driver is drowsy. However, from my perspective, it is tricky to set any threshold for the frequency as the frames labelled with yawning may be discontinuous. If someone is breathing deeply with mouth open, he is also likely to be evaluated as drowsy.

4.3 Future Work

In future work, we will try to improve the detection further by using an advanced method to capture the action on the driver's face. As there is another dataset containing drivers' frontal faces in YawDD, I can replace the current face images with them to achieve a better

recognition rate. To increase the accuracy of detecting the speaking, we can train the model by adding some specific features such as drivers' emotions. The samples from YawDD are not so accurate as well. Only a few seconds of content precisely describes the process of speaking in each video labelled with 'Speaking', which means the model receives many disturbing factors. That is the reason why the system cannot distinguish the normal and speaking efficiently. To handle the issue, we will increase the number of patients in this experiment and keep useful parts only. Also, one of the reasons that the prototype cannot be used widely is the ignorance of extreme conditions. When the driving occurs at night or on a cloudy day, the system may fail to capture the driver's face due to the lacking in lights. To deal with this problem, a model trained with night samples should be applied. Or we can add an infrared camera to the vehicle. Besides, enabling the prototype to run on a laptop is not practical in daily life. I will make efforts to enable the system run on a portable device that can indeed work in vehicles. Last but not least, as the restriction I mentioned above, I will improve the algorithm to focus on processing a series of images dynamically instead of analyzing the image one by one. Thus, the accuracy in real-time fatigue detection can be dramatically improved.

Reference

[1] Reddy, B., Kim, Y.-H., Yun, S., Seo, C., & Jang, J. Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[2] K. Zhang, Z. Zhang, Z. Li, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks", IEEE Signal Processing Letters, 2016.

[3] Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., & Jiang, S. Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques. Procedia Computer Science, 130, 400-407, 2018.

[4] Garcia, I., Bronte, S., Bergasa, L. M., Almazan, J., & Yebes, J. Vision-based drowsiness detector for real driving conditions. 2012 IEEE Intelligent Vehicles Symposium, 2012.

[5] J. F. May and C. L. Baldwin, "Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies," Transportation Research Part F: Traffic Psychology and Behaviour, vol. 12, no. 3, pp. 218 – 224, 2009.

[6] N. Kuamr and D. N. Barwar, "Analysis of Real Time Driver Fatigue Detection Based on Eye and Yawning," International Journal of Computer Science and Information Technologies, vol. 5 (6), pp. 7821–7826, 2014.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, 2004.

[8] Wu, Y.-S., Lee, T.-W., Wu, Q.-Z., & Liu, H.-S. An Eye State Recognition Method for Drowsiness Detection. 2010 IEEE 71st Vehicular Technology Conference, 2010.

[9] Zhang, W., Cheng, B., & Lin, Y. Driver drowsiness recognition based on computer vision technology. Tsinghua Science and Technology, 17(3), 354-362, 2012.

[10] Cootes T F, Taylor C J, Cooper D H, et al. Active shape models-their training and application. C*omputer Vision and Image Understanding*, 1995, 61(1): 38-59

[11] Sabet, M., Zoroofi, R. A., Sadeghniiat-Haghighi, K., & Sabbaghian, M. A new system for driver drowsiness and distraction detection. 20th Iranian Conference on Electrical Engineering (ICEE2012).

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-986, July 2002.

[13] Hachisuka, S. Human and Vehicle-Driver Drowsiness Detection by Facial Expression. 2013 International Conference on Biometrics and Kansei Engineering, 2013 [doi:10.1109/icbake.2013.89]

[14] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, No. 6, 2001, pp. 681-685.

[15] Vural, E., Çetin, M., Erçil, A., Littlewort, G., Bartlett, M., & Movellan, J. Machine Learning Systems for Detecting Driver Drowsiness. In-Vehicle Corpus and Signal Processing for Driver Behavior, 97–110, 2008 [doi:10.1007/978-0-387-79582-9_8]

[16] Tabrizi, P. R., & Zoroofi, R. A. Open/Closed Eye Analysis for Drowsiness Detection. 2008 First Workshops on Image Processing Theory, Tools and Applications, 2008. [doi:10.1109/ipta.2008.4743785]

[17] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, “YawDD: A Yawning Detection Dataset”, *Proc. ACM Multimedia Systems*, Singapore, March 19 -21 2014, pp. 24-28. [DOI: 10.1145/2557642.2563678]

[18] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.

[19] He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016 (CVPR). doi:10.1109/cvpr.2016.90

[20] Bradski G. The OpenCV Library. Dr Dobb's Journal of Software Tools, 2000.