SENIOR PROJECT

APPLICATION OF IMAGE PROCESSING TECHNIQUES ON VEHICLES

Submitted by:
Fahrulnisa Sema Özbaş
İbrahim Berber

Project Supervisor:
Asst. Prof. Shahram Taheri
Prof. Cafer Çalışkan

Faculty of Engineering
Antalya Bilim University
Spring 2020

TABLE OF CONTENTS

ABSTRACT 3
1. INTRODUCTION 4
1.1 Context and Motivation 4
1.2 Biometrics and Facial Recognition 6
1.3 Drowsiness Detection 7
1.4 Senior Project Purpose 7
2. LITERATURE REVIEW 8
2.1 Face Recognition 8
2.1.0 Biometrics point of view 8
2.1.1 Face Recognition Applications 9
2.1.2 Face Recognition Processes 10
2.1.3 Face Recognition Methods 11
2.2 CNN Architectures 14
2.3 Databases 16
2.4 Transfer Learning 17
2.5 Overfitting 18
2.5.1 Data Augmentation 18
2.5.2. Cross Validation 19
2.6 Drowsiness Detection 20
2.7 Viola – Jones Algorithm 21
3. PROJECT FRAMEWORK 22
4. IMPLEMENTATION DETAILS 23
4.0 Programming language 23
4.1 Face Recognition 23
4.1.1 Database selection 23
4.1.2 Architecture Selection 24
4.1.3 Database Preprocessing 25
4.1.4 Improvement Strategies 25
4.1.5 Parameter Selection 26
4.1.6 Image Acquisition and Processing 29
4.1.7 Algorithms for Face Recognition and Training Network 31
4.2 Drowsiness Detection 35
4.2.1 Algorithms for Eye Detection 35
4.2.2 Frame Acquisition and Eye Tracking Algorithm 35
4.2.3 Processing Methods 37
4.2.4 Evaluation Metrics 44
5. RESULTS AND EVALUTIONS 46
6. CONCLUSION AND FUTURE WORK 49
7. References 50

ABSTRACT

Automotive has an important place in human life, and the usage of personal vehicles is increasing. Unfortunately, this is also followed by the increase of car thefts and car accidents. Just in 2013 drowsy driving was responsible for more than 72,000 crashes and 700,288 motor vehicle thefts occurred. In order to prevent these situations, application of image processing techniques on vehicles are provided in our project.

The first goal of the project which is preventing vehicle theft, it is satisfied by creating face recognition system because of face recognition is providing a high level of security and high usability, it offers a non-intrusive, and perhaps the most natural, way of identification. System checks driver's face against a facial profile that already exists in the database which is linked to that person's file in order to find a match. According to the control result, the system confirms the driver and allows to access the vehicle, or rejects. The system is also allowing to register of new drivers. Because of having better performances in our system Convolutional Neural Network (CNN) is used which is a sub field of Deep Learning.

The second goal of the project, which is preventing drowsiness, is satisfied by creating real-time alert system. This detection of drowsiness is determined with eye status. For detection of eyes Viola-Jones algorithm is used which is algorithm of machine learning that allows the detection of image features in real-time. Since as drowsiness increases, eye openness will be decrease, in our project we measure the driver's drowsiness over openness level of his/her eyes.

Finally, it is hoped that this project would keep the car safe, prevent them from stealing and enhance driver's safety, prevent deaths and injuries.

Keywords – *Face Recognition, Machine Learning, Convolutional Neural Network, Deep Learning, Viola Jones*

1. INTRODUCTION

1.1 Context and Motivation

Automotive industry is growing each day and the usage of personal vehicles is increasing proportionally. Since 2016, 70,000,000 cars (or automobiles) which corresponds to 74% of the total motor vehicle, are produced each year. Unfortunately, this growth is also followed by the increase of car thefts and car accidents.

Vehicle Thefts

The average dollar loss per theft was $8,407. Vehicles were stolen at a rate of about 200 per 100,000 people in 2018, down from 230 in 2017. Following year, in 2018, 748,841 vehicles were stolen, and not decreased much, in 2017 which was 772,943 vehicles.

According to the data of National Insurance Crime Bureau, 765,484 motor vehicle thefts occurred in 2016 and in 2017, it has increased by 4.1%. In 2018, about $6 billion was lost to motor vehicle theft in the US alone. These imply the need for a new reliable safety measures that would keep the car safe and prevent them from stealing.

Year	Vehicles stolen	Percent change
2009	795,652	-17.0
2010	739,565	-7.0
2011	716,508	-3.1
2012	723,186	0.9
2013	700,288	-3.2
2014	686,803	-1.9
2015	713,063	3.8
2016	767,290	7.6
2017	772,943	0.7
2018	748,841	-3.1

Table 1.1. Motor Vehicle Theft, 2009-2018 (Source: U.S. Department of Justice)

Vehicle accidents caused by drowsiness

Another crucial problem related with vehicle is the accidents. Shocking statistics revealed by World Health Organization (WHO) in a 2009 report showed that more than 1.2 million people die on roads around the world every year. Moreover, an additional 20 to 50 million individuals suffer non-fatal injuries.

American Automobile Association released a figure in 2010 that 17% of all fatal crashes in the USA could be attributed to tired drivers. This seems to be a global trend. In Germany, several studies conducted by Volkswagen AG in 2005 indicate that 5-25% of all collisions are caused by driver falling asleep.

A survey conducted by National Sleep Foundation' Sleep in America 2009, 54% of the participants of the survey reported that they have driven at least once while drowsy in the past year. The National Highway Traffic Safety Administration estimates that drowsy driving was responsible for more than 70,000 crashes and caused the death of 800 in 2013, those numbers are claimed to be underestimated and 6,000 fatal crashes happens each year due to drowsy drivers. The Table 1.2 shows the number of total accidents, along with number of people died and injured from the years between 2009 and 2012 in Turkey.

Year	Total Accidents	Number of persons killed	Number of persons injured
2009	1 053 346	4323	201380
2010	1 106 201	4045	211496
2011	1 228 928	3835	238074
2012	1 296 634	3750	268079

Table 1.2. Total accidents, number of accidents involving death and personal injury, number of persons killed and injured between 2009 and 2012 in Turkey

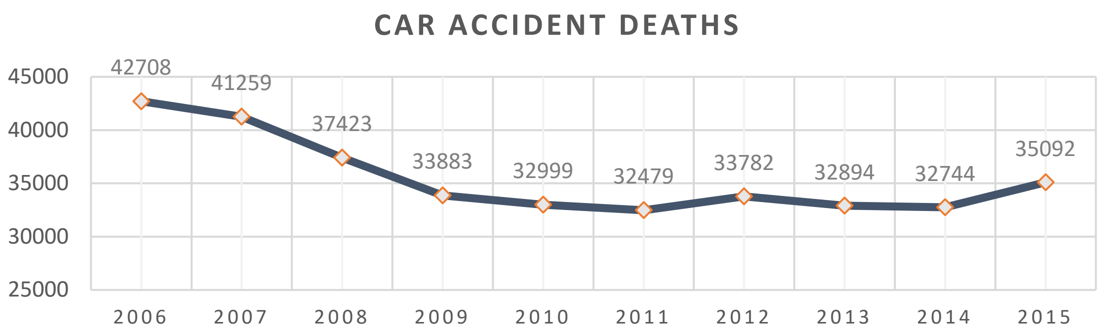


Figure 1.1. Number of car accident deaths over the years from 2006 to 2015, in USA

Fatigue related crashes are often more severe than others because driver's reaction times are delayed, or the drivers have failed to make any maneuvers to avoid a crash. As a remedy, there are safety technologies being employed and this is expected to make the cars smarter and significantly reduce the accidents caused by drowsiness of drivers.

1.2 Biometrics and Facial Recognition

In order to address vehicle thefts problem, biometric and non-biometric methods are implemented and usually provide such security features. From these implementations, non-biometrics measures sometimes have failed because of hacked password or unauthorized encryption of decrypted data, however, unlike the use of other forms of authentication, such as passwords or tokens, biometric recognition provides a strong link between an individual and access. This is because it is almost impossible to make replica of distinctive characteristics.

There are various biometrics verification approaches, such as fingerprint recognition, vein recognition, iris recognition and face recognition. Among them, the face is considered to be the most commonly used biometric trait. Considering application in vehicles which can be a remedy to problems related with protection of cars, face recognition is providing a high level of security and high usability, and it offers a non-intrusive, and perhaps the most natural, way of identification.

In fact, from industry drives perspective, according to The Global Government Biometric Systems Market 2015-2025, face recognition corresponds to about 29% of the biometrics market. The face recognition is predicted to have $10.2 billion by 2025.

Modality	Market Share Percentage
Fingerprint Recognition	34.9 %
Face Recognition	29.8 %
Iris/Retinal Recognition	14.9 %
Signature Recognition	7.8 %
Other	12.7 %

Table 1.3. Market share by biometric system in 2015

Face recognition addresses the problem of identifying or verifying one or more persons of interest in a scene by comparing input faces with face images stored in a database.

The important properties of face recognition are being non-intrusive and non-contact, and these makes facial recognition methods preferable in authentication in many areas, over other biometrics verification methods.

All these make using facial recognition for verification an appropriate alternative in driver authentication of vehicles, and the usage of face recognition in vehicle industry may provide a solution that reduces the number of thefts as there will be yet another additional security layer.

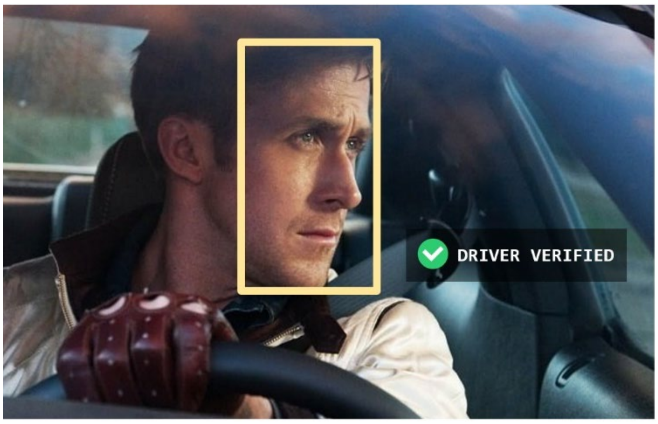


Figure 1.2. Identification of driver from his face

1.3 Drowsiness Detection

The level of drowsiness can be obtained by many approaches, however, one of simplest approach is a classification of driver based on his/her eye states, either having open or closed. Since as drowsiness increases, eye openness will be decrease, driver's drowsiness can be measured over openness level of his/her eyes.

Figure 1.3. Drowsy driver with closed eyes

By tracking the state of the face driver with classification methods, the categorization of driver as being drowsy or not, can be performed. Having this accomplished can lead to some other actions and security measures to be taken, e.g. warn or wake the driver, to decrease the number of deaths in car crashes caused by asleep during driving.

1.4 Senior Project Purpose

In this project, the goal is designing a system, which consists of two parts: Face recognition and driver drowsiness analysis. The system checks driver's face against a facial profile that already exists in the database which is linked to that person's file in order to find a match. The detection of drowsiness is determined with eye status.

2. LITERATURE REVIEW

2.1 Face Recognition

A human face reveals a great deal of information to a perceiver. This information is not limited by the mood and intention, but it includes a way of serving to identify a person. Although there are other biometrics attributes such as voice or fingerprint, the face is considered to be the most commonly used biometric trait by humans. Face recognition is one of the most important applications of biometrics-based authentication system, especially in last two decades.

2.1.0 Biometrics point of view

Face or facial recognition is identifying or verifying one or more persons in the scene by comparing with faces stored in a database, given static or video images of a scene. Face recognition relies on complex and dynamic structure of human face.

The problem of face recognition involves comparing two face images and determining if they are of the same person, and it is facing with these two challenges:

- Intra-user variations: Same person having different variations. (e.g. Figure 2.1)
- Inter-class similarities: Refers to the overlap of the biometric samples from two different individuals in the feature space. (e.g. Figure 2.2)

Figure 2.1. Intra-user problem example (pose, illumination and expression, respectively)

Figure 2.2. Inter-class problem example, similar facial appearance of twins

Identification and verification differ within the context of biometrics. Identification seeks to identify an unknown person, or unknown biometric. The system tries to answer the questions “Who is this person?”. Validation deals with “Is this person who they say they are?”.

2.1.1 Face Recognition Applications

With the popularity of face recognition, its applications have grown and continues to grow even more. Face recognition technology is well advance that can be applied for many commercial applications. The most widely used applications are presented in Table 2.1 below.

Area	Application
Security	Access control, flight boarding system, office access
Criminal Justice	Forensics and post-event analysis
Image Database Investigation	National ID, welfare registration, licensed drivers
Surveillance	Searching drug offenders, portal control
Video Indexing	Labeling the faces in videos
Human Computer	Interactive gaming and proactive computing

Table 2.1 – Face Recognition Applications (Barnouti 2016)

The applications can be grouped into two: identification and verification. In the identification problem, the face to be recognized is unknown and matched against face of a database containing known individuals. In the verification problem the system confirms or rejects the claimed identify of the input face.

Figure 2.3 – Facial Recognition Vehicle Access from Porsche

2.1.2 Face Recognition Processes

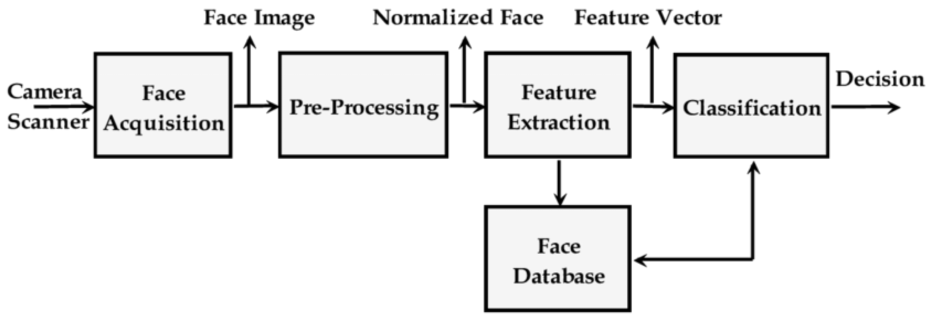


Figure 2.4 – Face Recognition Processes

The general flow is shown in Figure 2.4, above. In the beginning, the face image is acquired via camera or a scanner. Here, the face is detected. In other words, given an image, the question of "does it contain a human face?" is answered. Hence, face image is obtained.

The pre-processing step includes processing the image, which contains human face, in a way such that feature extraction can work better. This part can contain digital image processing techniques, including resizing, cropping, histogram equalization, filtering, scaling, reflecting, rotating, and enhancement.

Facial feature extraction is the process of extracting face component features like eyes, nose, mouth, etc. from human face image. Facial feature extraction is very much important for the initialization of processing techniques like face tracking, facial expression recognition or face recognition. The extracted features then stored inside feature vectors.

Face database is the place where feature vectors are stored for need of later comparison when a test image is presented. For each individual, different feature are extracted and stored inside the database in an organized manner.

Classification step uses the comparison of features extracted from input image and the feature stored in database. In short, performs a matching of the face against one or more known faces in a prepared database.

Depending on the matching score, the system produces a decision. An example within the framework of authentication of vehicle is that, deciding whether the input face is authorized or not.

2.1.3 Face Recognition Methods

Principle Component Analysis (PCA)

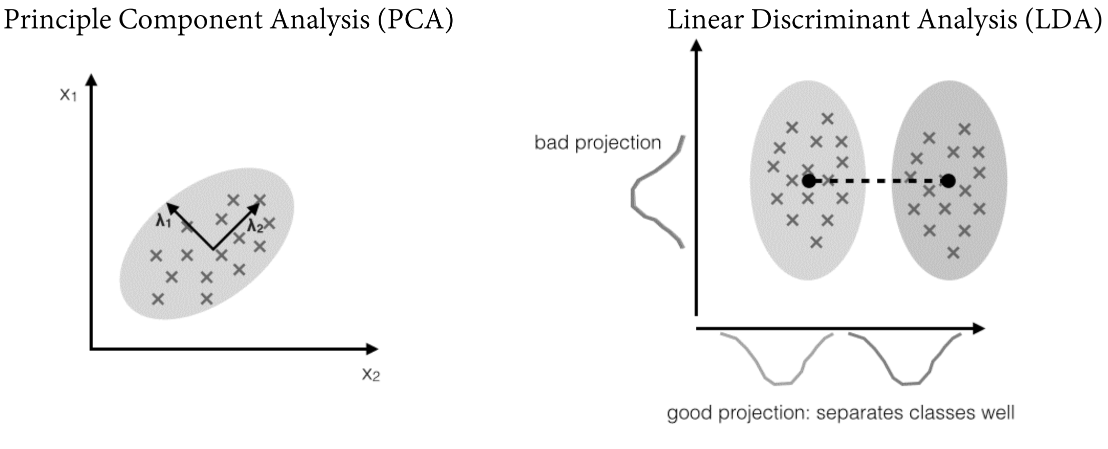
PCA is feature extraction and dimension reduction method. It can be used to solve the problems related with recognition and compression. PCA is a popular linear projection method. In this approach, the faces are transformed into smaller set of essential eigenvectors, which are named eigenfaces. These basis vectors are called principal components. PCA can extract the important features, capture the closely variable data components of samples, and then select several important individuals from all the feature components.

Linear Discriminant Analysis (LDA)

LDA is used for feature extraction and dimension reduction, its widely used to search for linear groups of features while maintaining the separate of classes and known as Fisherface which is the output of this method.

The face contains several features which are distinctive, and the goal of LDA is to reduce the number of features to a manageable number before the classification and maximize the ratio of the between-class variance, and the within-class variance (Gavrilova & Monwar, 2013).

LDA is an alternative for PCA and Experiments have shown 16 that LDA has a better performance and outperforms PCA when the number of samples per class is small (Martínez & Kak, 2001).



(a) (b)

Figure 2.5 – Differences of PCA (a) and LDA (b)

Local Binary Patterns (LBP)

Local binary patterns (LBP in short) is a type of visual descriptor used for classification in computer vision. LBP is also used in face recognition. It is a texture-based algorithm. Here, the texture is significant for determining characteristics of face image. LBP uses the facial feature extraction by dividing it into several areas which are relatively smaller. The term "operator" in Figure 2.6, refers to a mapping function that can transform an image from one form to another.

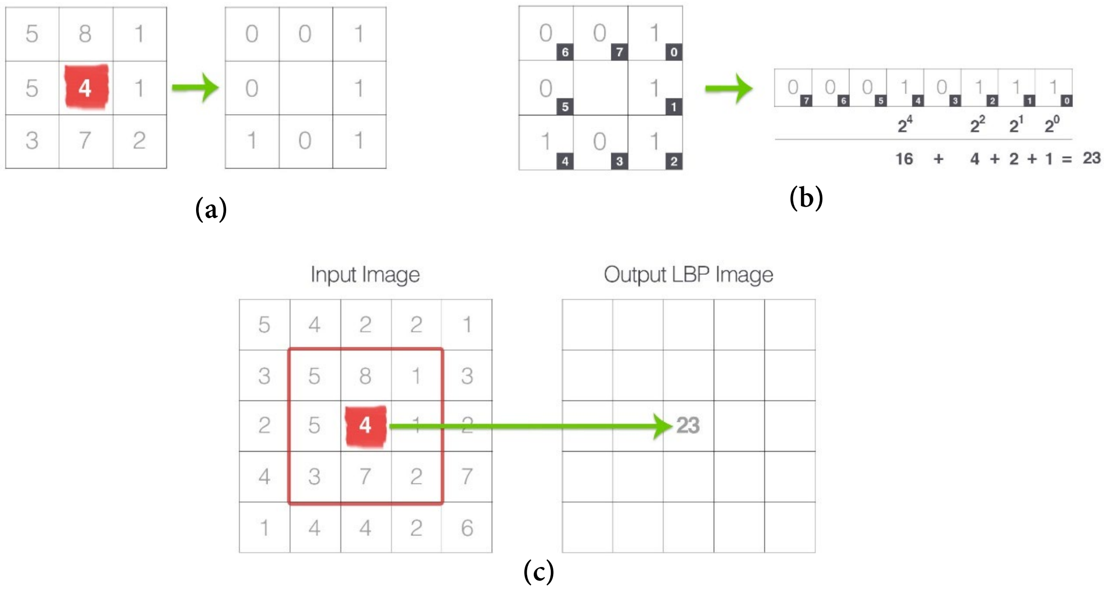


Figure 2.6 – Illustration of LBP Operator

Support Vector Machine (SVM)

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. Support Vector Machine use a training set of images in order to have the computation of optimal separation of hyper plane. Here, the separation is adapted to solve the classification problem which can be stated as finding correct face among given dataset.

Histograms of oriented gradients (HOG)

Histograms of oriented gradients (HOG) descriptor, which is inherited from scale invariant feature transform, performs quiet well on the human detection. In face recognition, the global and local features of images are often combined to use and achieve better results. Similarly, global and local HOG features have also been tried to be combined to use in shape classification and object detection and achieve better results.

Convolutional Neural Networks (CNN)

Neural networks are used to recognize the face through learning correct classification of the coefficients calculated by the eigenface algorithm. The network is first trained on the pictures from the face database, and then it is used to identify the face pictures given to it.

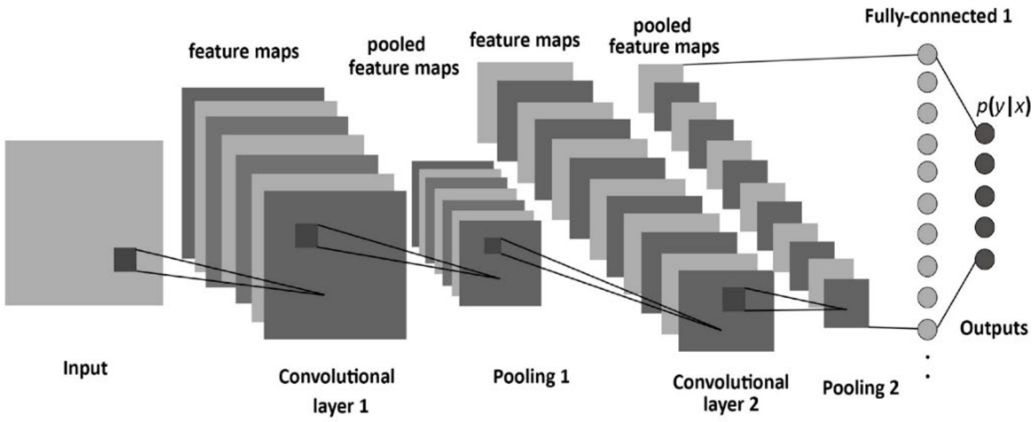


Figure 2.7 – Architecture of CNN

2.2 CNN Architectures

In the last five years, there are numerous CNN architectures are proposed, addressing various purposes. Some architectures whose performance is well and that is proven, are preferred in classification tasks in datasets such as COCO (Common Object in Context) or ImageNet. The used architectures are suitable for various tasks, as classification of images can be considered to cover any general task related to feature extraction from images.

2.2.1 AlexNet

AlexNet architecture has won the ImageNet Visual Recognition Challenge in 2012 and started the Deep Learning revolution. AlexNet was trained on ImageNet which has 1.2M images from 1000 classes. Here, Dropout is used to prevent overfitting.

2.2.2 VGG

Another deep convolutional neural network proposed is VGG net, which performs very good in ImageNet Challenge in 2014. In this architecture, 3x3 convolutional filters are used to improve accuracy.

Figure 2.8 – VGG Face Architecture

2.2.3 GoogLeNet

GoogLeNet is a convolutional neural network that is 22 layers deep. The trained version of network over ImageNet and Places365 is available. The trained model consisted of 1000 images including keyboard, mouse, pencil, and many animals.

2.2.4 ResNet

ResNet architecture was the winner of ILSVRC 2015 challenge, where the task was classification. Compared to VGG, ResNet is deeper but lower complexity. The core idea of ResNet is having “identity shortcut connection” which skips one or more layers.

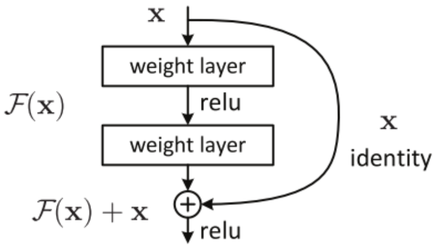


Figure 2.9 – A residual block in ResNet

2.2.5 Inception-v3

Inception-v3 is a convolutional neural network that has 48 layers. This architecture is used to classify to images. The computation cost is slightly higher than GoogLeNet, but much more efficient than VGGNet.

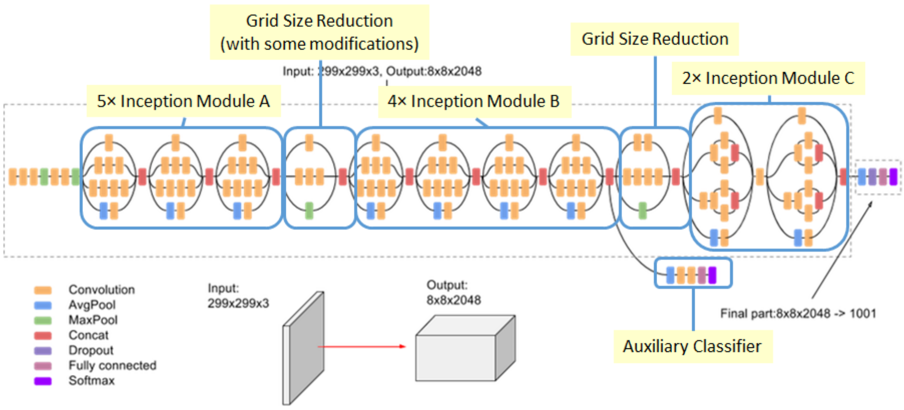


Figure 2.10 – The Inception-v3 architecture (Batch norm and ReLU are used after Conv)

2.3 Databases

There are several databases which designed for face recognition systems, available. Among them, the popular face databases are shown in Table 2.2. these datasets differ from each other by having different size, scope or purpose.

Considering the vehicle application RGB images could be better alternative.

Name	Individuals	Image Resolution
AT&T (ORL)	40	92 x 112 (Gray Scale)
FERET	1199	256 x 384 (Gray Scale/ RGB)
AR Face Database	126 (70 M, 56 F)	576 x 768 (RGB)
PIE Database, CMU	68	640 x 486 (RGB)
BioID Face Database	23	382 x 288 (Gray Scale)
The Yale Face Database	15 (14 F, 1 M)	320 x 243 (Gray Scale)
The Yale Face Database B	10	640 x 480 (Gray Scale)
UMIST Face Database	20	92 x 112 (Gray Scale)
The MUCT Landmarked	276	480 x 640 (RGB)
Labeled Faces in the Wild	5749	250 x 250 (RGB)
Faces96	151	196x196 (RGB)
Indian Database	40	640 x 480 (RGB)
FEI Database	200	640 x 480 (RGB)

Table 2.2 – Commonly used face databases

Figure 2.11 – Example photos from Labeled Faces in the Wild

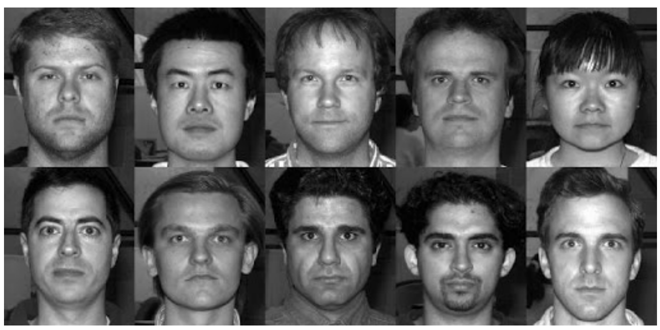


Figure 2.12 – Example photos from The Yale Face Database

2.4 Transfer Learning

Transfer learning refers to reusing the knowledge learned from one task for another. Specifically, for convolutional neural networks (CNNs), many image features are common to a variety of datasets. These features include lines, edges which can be seen in almost every image. Since training an entire CNN from scratch with a random initialization, requires more computation power and takes so much time, Transfer Learning methods are performed. In large structures, it is rare to train a CNN from scratch because of computational requirements.

Depending on the size of the dataset and similarity of the data in dataset, transfer learning has two approaches.

If the new dataset is almost similar to the original dataset used for transfer learning, the same weights can be used for extracting the features from the new dataset.

If the new dataset is very small, it can be training only the final layers of the network to avoid overfitting, keeping all other layers fixed. For opposite situation, if the new dataset is very much large, retrain the whole network with initial weights from the pretrained model.

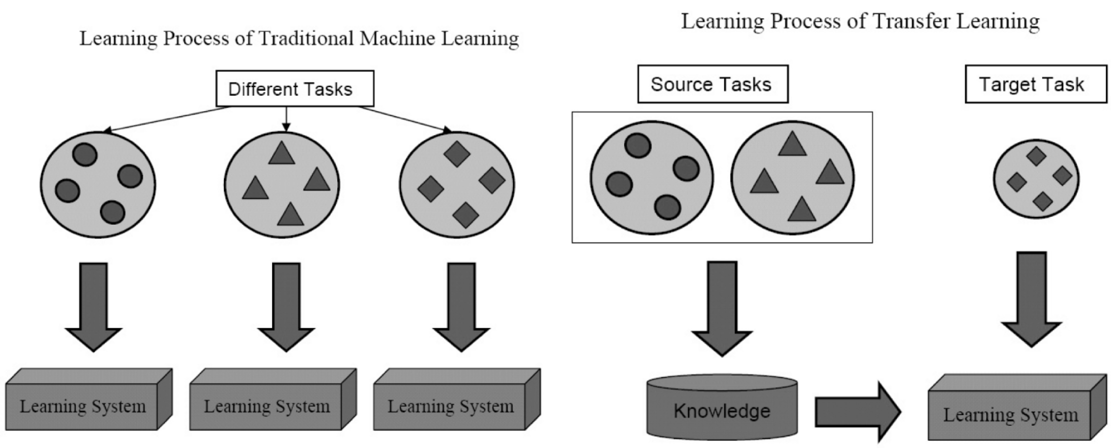


Figure 2.13 – Different Learning Processes between Traditional Machine Learning and Transfer Learning

2.5 Overfitting

Overfitting is a common problem in machine learning and data science – a risk when training neural networks is that they might get overfitted. A model is overfitted when feeding it with a lot of data more than necessary. When a model fits more data than it actually needs, it starts catching the noisy data and inaccurate values in the data. As a result of, accuracy of the model decrease. This means that although the model fits the training data very well, it does not generalize to other data. There are several methods of preventing this from happening, two of which are described below, cross-validation, early stopping, data augmentation.

2.5.1 Data Augmentation

Neural networks require a large quantity of data when training, in order to produce good results. Collecting such large datasets is not always feasible, due to lack of time or resources. One way of getting around this issue is by expanding smaller datasets, using different transformations or other kinds of processing, data augmentation is described as a way of preventing overfitting. Furthermore, in the real-world scenario, may have a dataset of images taken in a limited set of conditions. But target application may exist in a variety of conditions, such as different orientation, location, scale, brightness etc. Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks.

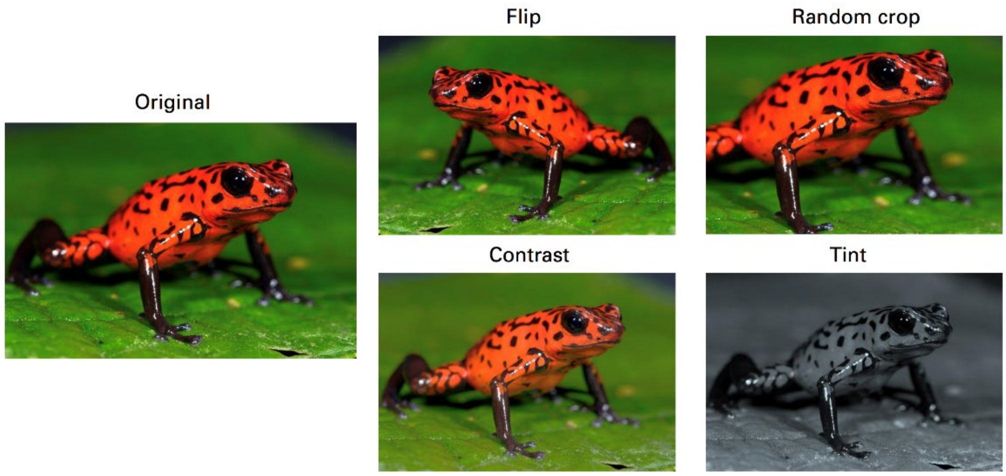


Figure 2.14 – Data augmentation examples

2.5.2. Cross Validation

Cross-validation is one of the most widely used data resampling methods to assess the generalization ability of a predictive model and to prevent overfitting. K-Fold Cross Validation is where a given data set is split into a K number of folds where each fold is used as a testing set at some point. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the K folds have been used as the testing set.

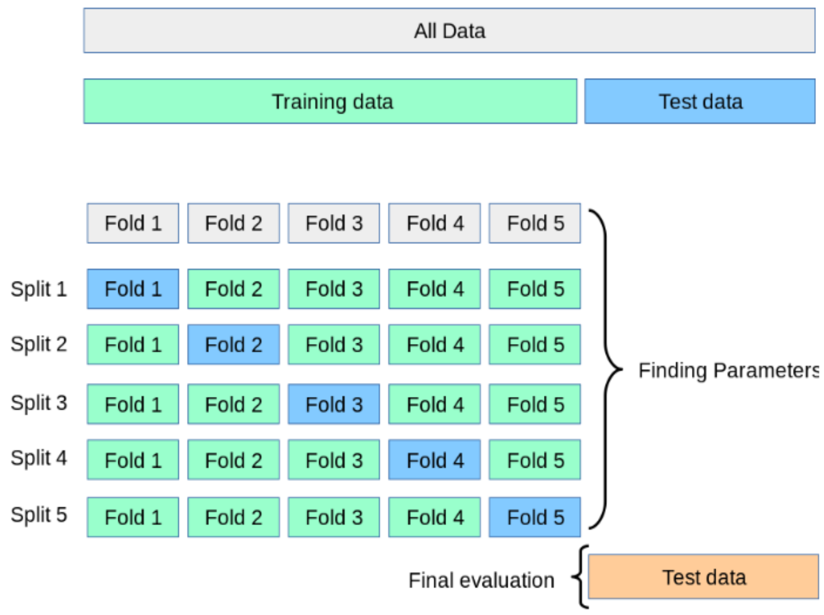


Figure 2.15 – K-fold Cross Validation

2.6 Drowsiness Detection

Driver being drowsy and fatigued is one of the situations commonly encountered in control of motor vehicles. Driving in drowsy state is a major cause leading to road accidents and driver exposes to much danger of crashing compared to alerted state.

There is various way of measuring the drowsiness level, as seen in Table 2.3, below. Many studies rely on the measurement done via eye condition. Figure 2.16 illustrates the states of eyes, which in drowsiness condition, the closure percentage increases.

Figure 2.16. Examples of different states of eye opening

Drowsiness Measure	Detection Techniques	Feature Extraction	Classification
PERCLOS, AECS, Gaze, Facial Expression	SVM Classifier, Kalman filter	Gabor wavelet regression neural network	BN
PERCLOS, AECS, Mouth opening	Viola-Jones Algorithm	Correlation coefficient template matching	SVM
Head pose, Yawning	Geometric Method	HOG, Haar features	SVM
PERCLOS	SVM Classifier, Harr features	HOG, Spectral Regression	Threshold based
PERCLOS	Haar-like features and Kalman filter	Local Binary Pattern (LBP)	SVM
Eye closure duration and freq.	Hough Transform	DWT	Neural Classifier
PERCLOS	Haar Algorithm	Kalman Filter	SVM
Yawning	Viola-Jones Algorithm	Viola-Jones Algorithm	SVM

Table 2.3. Ways of measuring drowsiness in literature

2.7 Viola – Jones Algorithm

The Viola-Jones algorithm is a prevalent used mechanism that allows the detection of image features in real-time described by Paul Viola and Michael Jones. The Viola-Jones algorithm has 2 important steps Haar-like features, classifier cascades, with these steps machine is trained for detecting objects in a picture.

The Haar-like features is a scalar product between the image and some Haar-like templates which is split into 2-4 number of rectangles. Figure 17 shows the possible features and the Figure 18 shows the horizontal features describe the eyebrows, nose and mouth.

Figure 2.17. Haar-like templates

It is clear that the eyebrows are darker than the forehead and the middle of the nose is bright than two side of nose. Features can be utilized different ways, two rectangular regions which is white and black, the difference between the sum of the rectangular regions are calculated, three rectangular regions which are one white and two black, the difference between the sum of the two sides of rectangles and central rectangular region. All these features used to find a pattern to represent a face.

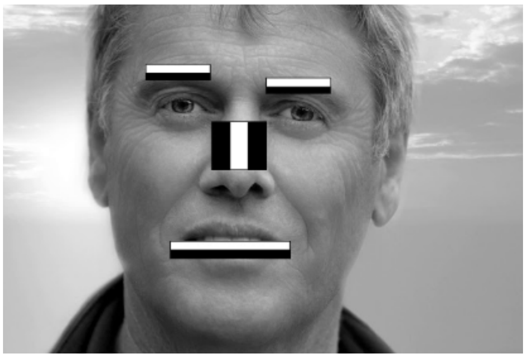


Figure 2.18. Haar-like templates

After evaluation of Haar-like templates, integral of image calculated., to get the sum of values in a rectangular area. For face detection, parts of the image that do not contain valuable information must be ignored and face should be focused. For this problem cascade classifier is used and to detect an image with faces completes successfully

3. PROJECT FRAMEWORK

The general workflow can be seen in Figure 3.1, below. Mainly, the project consists of two parts: facial recognition (Figure 3.1.a-c) and drowsiness detection (Figure 3.1.d and 3.1.e) based on eye status.

Figure 3.1.a shows the capture and registration process of an image acquired by hardware. Then, the architecture is trained with database which includes the registered driver, as illustrated Figure 3.1.b. Here, the trained model is stored for later use. The classification of input image, acquired by camera, happens in Figure 3.1.c. Granted or rejected driver will be displayed as genuine or imposter, respectively. Open eye template image is registered in Figure 3.1.d, employing image processing techniques. Last, but not least, Figure 3.1.e illustrates the process flow regarding the detection of drowsiness. Detected eye pair is monitored continuously and closed eye give rises to an alert.

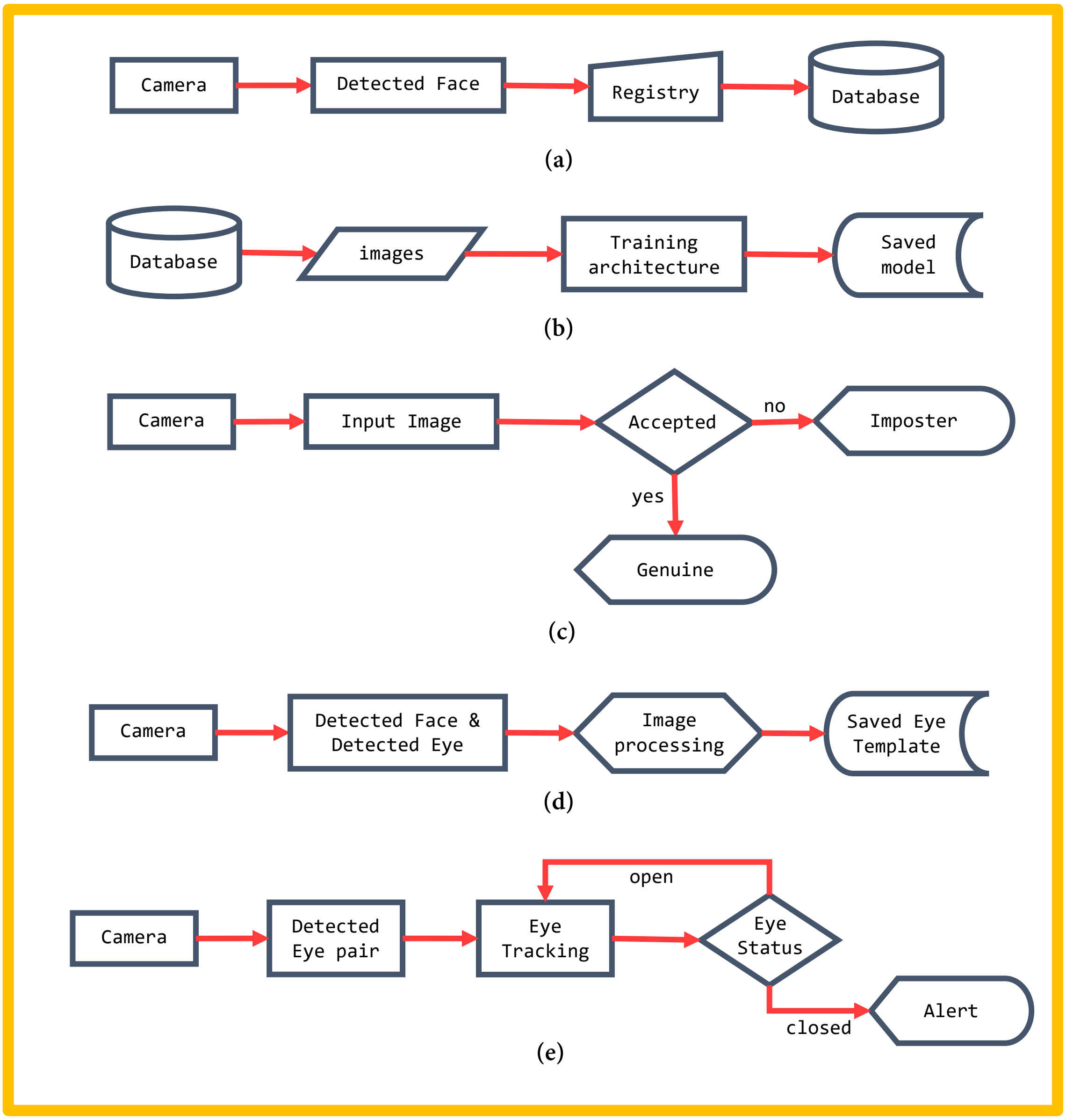


Figure 3.1. Flow Chart of Project Progresses

4. IMPLEMENTATION DETAILS

4.0 Programming language

The selected programming language was determined according to the project criteria and advisor recommendations. As a result of the researches, it is understood that MATLAB coding environment is a user-friendly programming language that involves a variety of libraries related with deep learning and image processing, it is fast, and has an acceptable performance. After research, MATLAB was selected for this project.

4.1 Face Recognition

4.1.1 Database selection

Selection of database is significant in order to obtain a desired result. Regarding the purpose of the project, we have considered various datasets, (as mentioned in Section 2.3). Databases which consist of images that have higher resolution and RGB colored are preferable to get better results.

Labeled Faces in the Wild has images of 5749 individuals, with each image having 250 x 250 dimensions (RGB). However, number of images for each individual are not consistent in this database. While, for one individual, there exists more than 10 images, for some other individual, only 1 image exists. For recognition task with CNN, the higher number of instances for each individual gives better accuracy.

The other dataset we have taken into consideration is Faces96. This data provides 20 images for each individual. However, we have considered that having resolution of 196 x 196 could be low in practical usage. Additionally, the 20 images were very similar to each other. In other words, the subject was staring at the camera with a constant angle, which reduces the performance of the recognition.

In real-life situations, having images taken from various angles and different illumination level are more likely. Hence, FEI dataset is appropriate choice. The FEI face database used in this study is a Brazilian face database containing a set of face images and which researchers can use in their projects. In the dataset, there are 200 different young and middle-aged individuals and each image taken with 14 various perspective such as different orientation, brightness etc., as shown in Figure 4.1. All images are RGB and have white background with original size of each image is 640x480 pixels.

Figure 4.1. Examples of various perspective images from one individual in the FEI face

4.1.2 Architecture Selection

After deciding dataset, next selection is the architecture. From literature, many alternatives exist. We have tried AlexNet, SqueezeNet, ResNet-18, VGG-16 and Inception-v3. Considering computational requirements and accuracy-wise comparison, Inceptionv3 was selected because it gives more optimum results than other pretrained neural networks that we have tried on our selected dataset in this study. Our findings are in correlation with what is shown in Figure 4.2, which shows comparison the ImageNet validation accuracy with the time required to make a prediction using the network

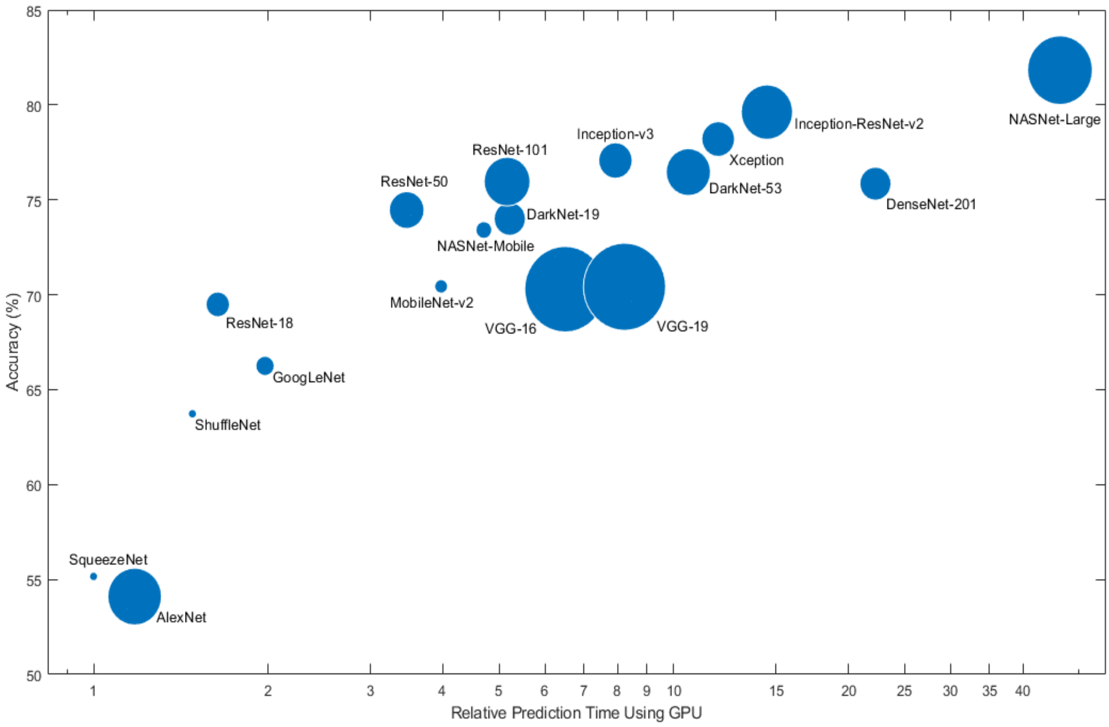


Figure 4.2. Comparison ImageNet accuracy vs required time in prediction of networks

Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	4.6 MB	1.24	227-by-227
googlenet	22	27 MB	7.0	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20.0	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224

Figure 4.3. Available pretrained network on MATLAB

Source (https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html)

4.1.3 Database Preprocessing

The dataset in original form contains images in one root folder. For the purpose of classification, we turned them into separate folders where each folder corresponds to one individual. As seen in the Figure 4.4, the last image for each person is taken with lower illuminance level. As we trained to model, we notice that reduces the test accuracy. Hence, the last image (14^{th}) of each individual is taken out. Additionally, we reserved the 11^{th} image for later use, which is testing. Consequently, the training will be on images from 1 to 13 except 11^{th}, and the model will be tested with an image that the system had never encounter with before. We also extended the database with our 14 images.

```
Algorithm MakeFolders
construct dictionary for each individual
create separate folders
for each individual
    take out last image
    reserve 11th image in separate folders
```

Figure 4.4. An excluded image (low illuminance)

4.1.4 Improvement Strategies

The training performed with **`inceptionv3`** CNN architecture. The *Inception-v3 Network* is available on **`MATLAB`** with Deep Learning Toolbox™. For training options, there are various parameters which we can tune. In order to have improvement, we use data augmentation and cross-validation.

In this study, **`imageDataAugmenter`** is used to specify additional augmentation operations to perform on the training images to help prevent the network from overfitting. For cross-validation, the *k*-fold size is selected as 5. This will guarantee that each 20% of our dataset will be used on training. The average of five training is computed to determine overall accuracy.

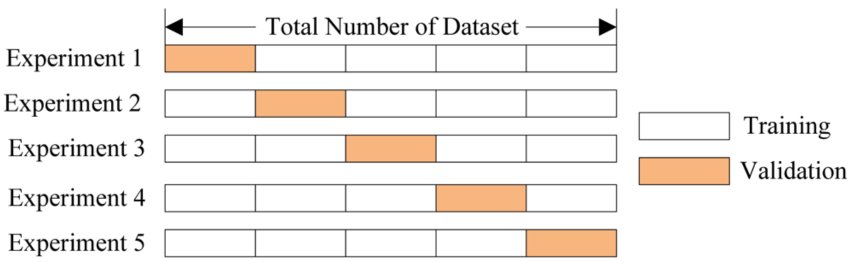


Figure 4.5. Cross-validation (source: Kaggle)

4.1.5 Parameter Selection

The parameters are selected based on exhaustive search of best possible combination. Some of which we have tried are presented in Table 4.1, below. In all setups, 5-*fold* was performed and the average of these five accuracies are taken. Not surprisingly, having higher Epoch allowed us to obtain higher accuracies. The learning rate of 1e-4, in general, was too low to converge. Considering the time required for training, having mini-match size of 16, with learning rate of 1e-3 was an appropriate choice. In some cases (e.g. shown with * in table), we stopped training because of having very low accuracies in the first fold. Some accuracy plots are shown on next two pages.

Database	Learning rate	Mini Batch size	Epoch	Avg. Accuracy
Faces96	1e-3	16	2	99.73 %
Faces96	1e-4	16	2	55.63* %
Faces96	1e-4	32	2	18.83 %
Faces96	1e-3	16	3	99.87 %
FEI Database Part 1	1e-3	16	2	95.73 %
FEI Database Part 1	1e-3	32	2	91.20 %
FEI Database Part 1	1e-4	16	2	34.27 %
FEI Database Part 1	1e-4	32	2	**11.00** %
FEI Database Part 1	1e-3	16	3	97.67 %
FEI Database Part 1	1e-3	32	4	96.47 %
FEI Database Part 1	1e-4	16	4	73.80 %
FEI Database Part 1	1e-4	16	8	93.60 %
FEI Database Part 1	1e-4	32	8	81.67 %
FEI Database Part 1	1e-3	16	8	97.67 %
FEI Database	1e-3	16	8	98.37 %
FEI Database	1e-3	16	2	96.90 %
FEI Database_v1	1e-3	16	2	98.10 %
FEI Database_v2	1e-3	16	2	97.70 %
FEI Database_v3	1e-3	16	2	97.86 %
FEI Database_v4	1e-2	32	1	85.07 %
FEI Database_v4	1e-3	32	4	98.68 %
FEI Database_v4	1e-3	16	4	99.70 %
FEI Database_v4	5e-3	16	4	98.83 %
FEI Database_v4	1e-3	16	3	99.50 %
FEI Database_v4	1e-4	16	3	58.17 %
FEI Database_v4	1e-3	32	3	99.10 %

Table 4.1. The experimental results we have performed with various combinations of parameters; *FEI Database Part 1* → 50 people; FEI Database_v1 → 14th image excluded and 13th is test image; FEI Database_v2 → 14th image excluded 11th is test image; FEI Database_v3 → *Sema* included; FEI Database_v4→ *Sema* and *Ibrahim* included

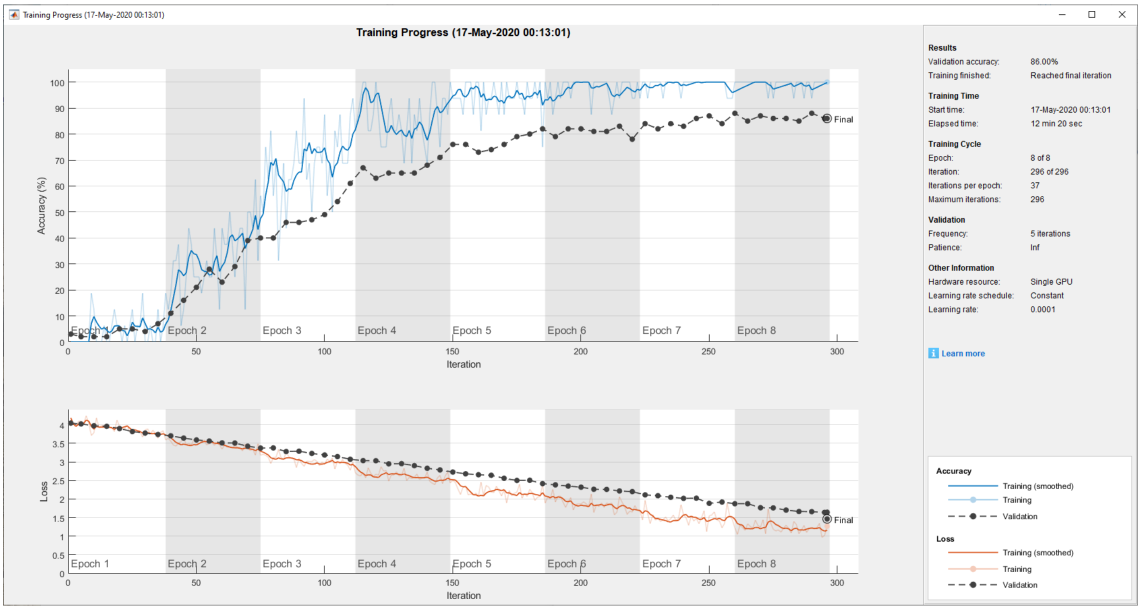


Figure 4.6. Epoch = 8, Learning rate = 1E-4, Mini-Batch size = 16, *FEI_Dataset*
(Possible overfitting, there a gap in validation and trained datapoints)

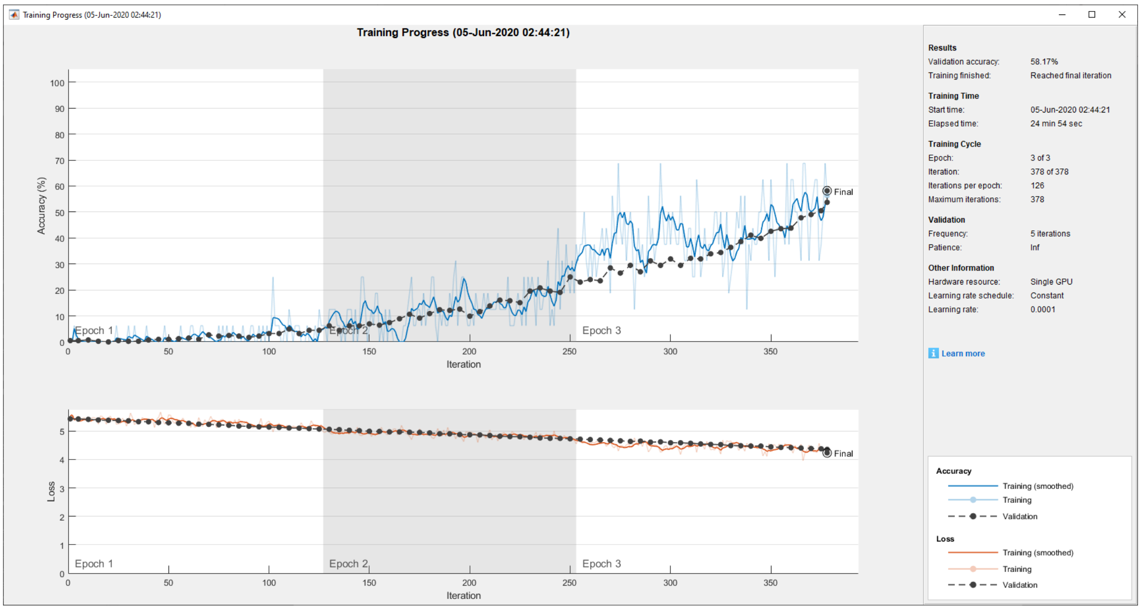


Figure 4.7. Epoch = 3, Learning rate = 1E-4, Mini-Batch size = 16, *FEI_Dataset*
(Too much fluctuate and not learning rate is too slow)

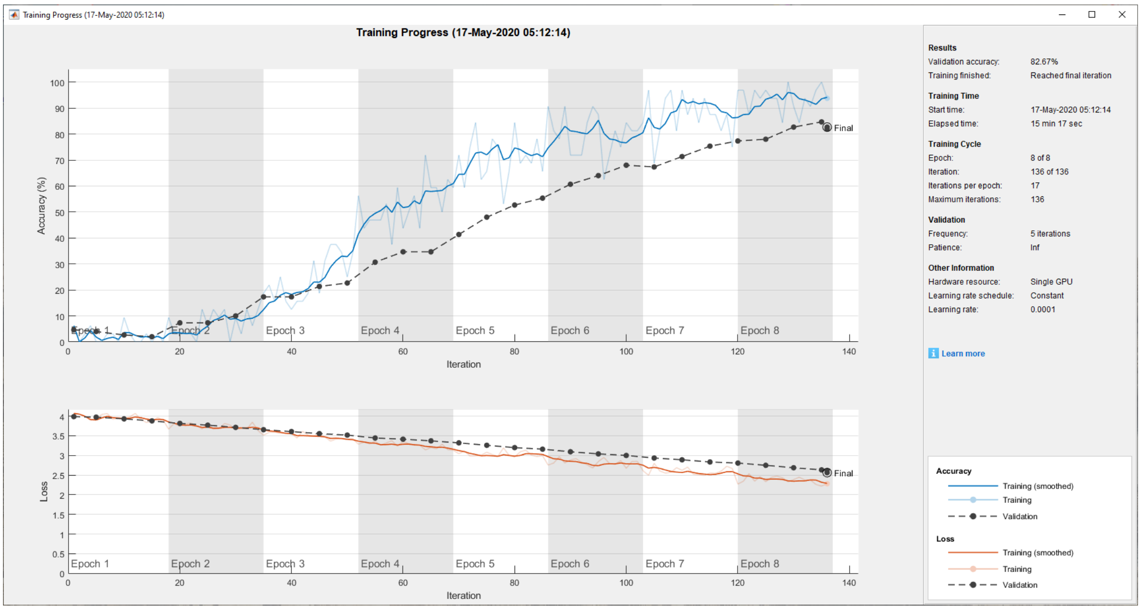


Figure 4.8. Epoch = 8, Learning rate = 1E-4, Mini-Batch size = 32, *FEI_Dataset* (Not converges completely, and possible overfitting)

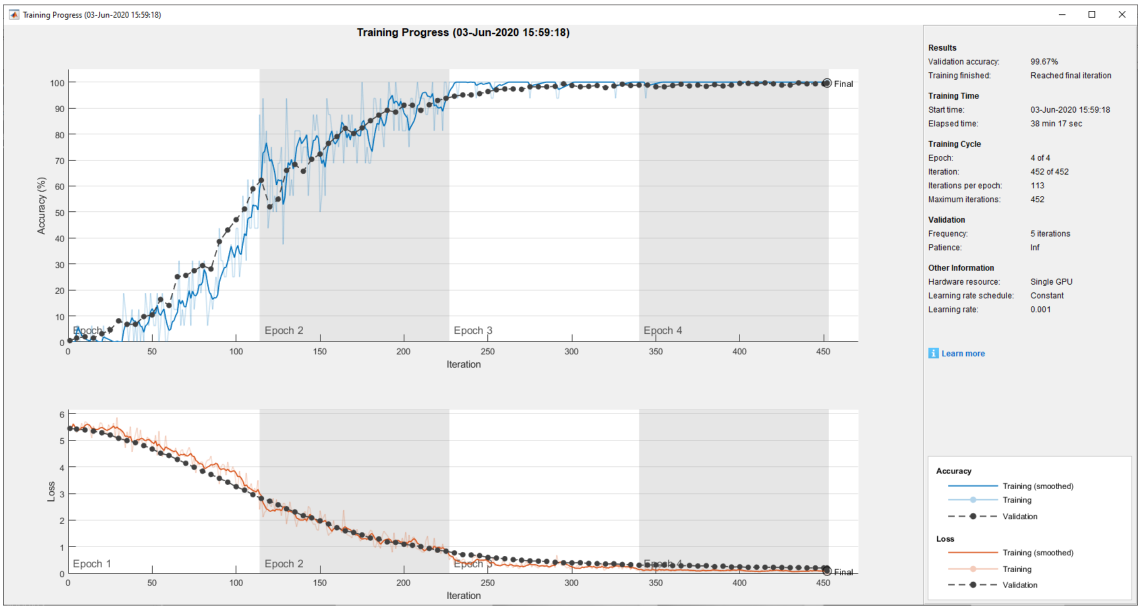


Figure 4.9. Epoch = 4, Learning rate = 1E-3, Mini-Batch size = 16, *FEI_Dataset_v4* (Our optimum selected parameters)

4.1.6 Image Acquisition and Processing

The images of subject (driver) can be registered using **`Register Driver`** GUI Application, as seen in Figure 10, below. The program relies on **`Image Acquisition Toolbox`** in **`MATLAB`**.

After the program executes, user has to write his/her name into Driver name textbox. This will enable him to start the camera activity. User can activate the camera using **`Activate camera`** button. The program will detect the installed hardware camera on the system. In our case, the images will be captured using laptop computer webcam (see Figure 10), however, it is also possible to capture the images via some other external camera (e.g. raspberry pi camera device).

When the camera is activated, real-time image stream will be displayed on the program. Here, the face is detected in real-time, using Viola-Jones algorithm. In **`MATLAB`**, **`vision.CascadeObjectDetector`** is used, (available in **`Computer Vision Toolbox`**) which is built on Viola-Jones algorithm, with default parameters. The default parameters execute for face detection. The detected face is annotated with **`Image Labeler`**, and it is surrounded by rectangle with annotation (see Figure 10).

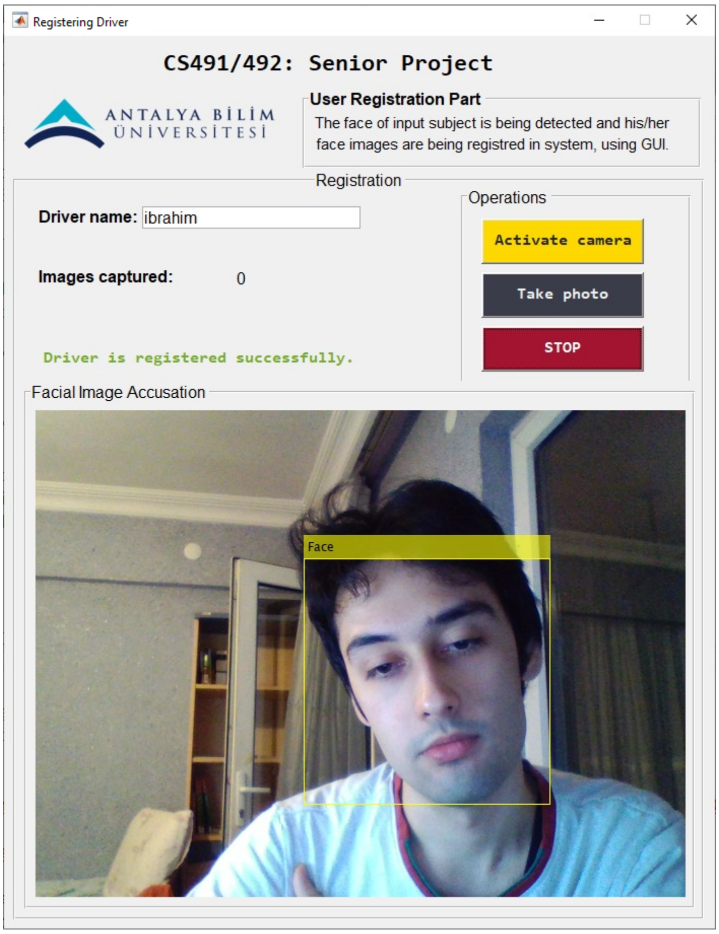


Figure 4.10. Registration GUI

Figure 4.11. External hardware device example

```
Command Window
cam =

  webcam with properties:

                    Name: 'BisonCam, NB Pro'
    AvailableResolutions: {'640x480'  '160x120'  '176x144'  '320x240'  '352x288'  '640x360'  '1280x720'  '1280x1024'  '1920x1080'}
              Resolution: '1920x1080'
              Saturation: 50
                Contrast: 50
        WhiteBalanceMode: 'auto'
               Sharpness: 50
              Brightness: 50
                     Hue: 50
            WhiteBalance: 4500
                   Gamma: 50
```

Figure 4.12. Specs of camera on our laptop computer

In live-stream, user can take his/her photo by clicking on **`Take photo`** button after he/she make sure that his/her face is detected and annotated. The captured images are stored in a folder which is named as the Driver name. For our selected database, it is recommended to take 14 images. The captured images have the resolution of 640 x 480 pixels, same as with dimensions of images in selected database.

Once registration is desired to be completed, the program should be terminated by clicking on **`STOP`**. This will release the camera objects in **`MATLAB`**, otherwise the error will occur. By doing this, registration process is done.

The images stored in folders are used for either training or test purposes later. In our project, we combine these images with the images collected from volunteers (i.e. selected face dataset).

4.1.7 Algorithms for Face Recognition and Training Network

Given a database, the files in each folder are read by **`imageDatastore`** in **`MATLAB`**. The use of **`ImageDatastore`** object helps us to manage a collection of image files, where each individual image fits in memory. Then the images are partitioned into five in randomized manner, because the k-fold was selected to 5. Having cross validation ***k*** of 5, guaranties that each portion of dataset will be used as training.

Then, the training of selected architecture, which is inception-v3, begins. Since each architecture has different layer design, the **`Fully-Connected Layer`** (FC-Layer) and **`classificationLayer`** are modified for our database. For our purpose, we wanted to train a network which will be able to recognize 202 different classes (200 individuals from FEI database and our 2 group members). Consequently, number of classes in our training network **`numClasses`** is 202. Therefore, we replace the original FC-Layer in a way such that it will recognize our **`numClasses`** which 202, and with this, we have incorporated our dataset into training process. Obviously, new users have to register their images to the database and train the network accordingly.

During the training, to have closer to real-life situations and obtain better accuracies the data augmentation was performed. For each fold, 80% of data is selected and 20% of data used as test, each being unique. In order to see how well it was trained, the predictions, which corresponds to test data, are compared with actual validation data. The obtained accuracies are displayed. After each ***k***-fold training, the accuracies are observed, and desired training setup was saved for later use in recognition task.

```
accuracy =

    0.9725    0.9850    0.9733    0.9825    0.9717

The System Accuracy is 97.70
fx >>                                     teams.microsoft.com is
```

Figure 4.13. Trained model accuracies with 5-fold and average

For recognition to work, the network must be trained previously with driver's face. Once trained network is stored in system (for our case local computer), it is ready for actual recognition task with a provided image. In testing, the image can be given in many ways, including even the Registration GUI.

Since we want to authorize not all trained individuals, but the specified users, we select them with another GUI shown in Figure 4.14 on the right. In other words, rather than *recognizing* 202 individuals, the system will accept the ones we have instructed the program to do so. The simple GUI creates a file where the authorized individuals (which has vehicle access are provided) are stored. Also, authorized ones can be removed. The file then will be used for recognition testing.

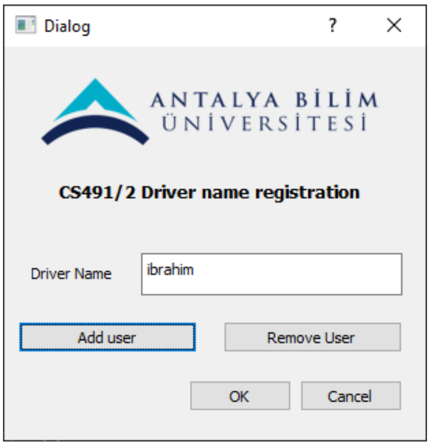


Figure 4.14. Authorized users selected

The model, which is trained with images including driver's face, is loaded. So, the model is not trained over again and used directly. Then, test image is loaded. Depending on the matching percentages of given input image, top 5 predictions are displayed. The strongest prediction is decided to be predicted individual.

The next step is deciding whether the predicted individual is an authorized one. If prediction subject is stored in authorized database, then the system grants access to that individual. For system to classify given subject genuine, the prediction percentage should be greater than or equal to 40%. The prediction percentage threshold value can (and should) be varied depending on the hardware used to capture the image. The more similar format to registered image, is followed, the better performance can be obtained. If the top predicted user has the prediction score of lower than specified threshold, but top predicted individual is in the authorized list, it will still be rejected.

Algorithm ProvideAccess

load authorized list
load trained model
read subject image
if top predicted user **in** authorized list **and** prediction score >= 40:
    provide access
else:
    access denied

There are two conditions in which the algorithm should behave appropriately. Registered driver's face should be recognized, and unregistered ones should be classified as imposter, meaning their access will be denied. In our extended database (200 individuals and images of our group members), we have registered our names with a person in data base ('sema', 'ibrahim', '11') as authorized user, and tested some variations to validate the working of access providing algorithm.

In validation test we have done, firstly, predetermined users who have authorized are detected with impressive accuracy as seen in Figure 15 and 16, above. The strongest prediction has an outstandingly high percentage compared to following four prediction percentages.

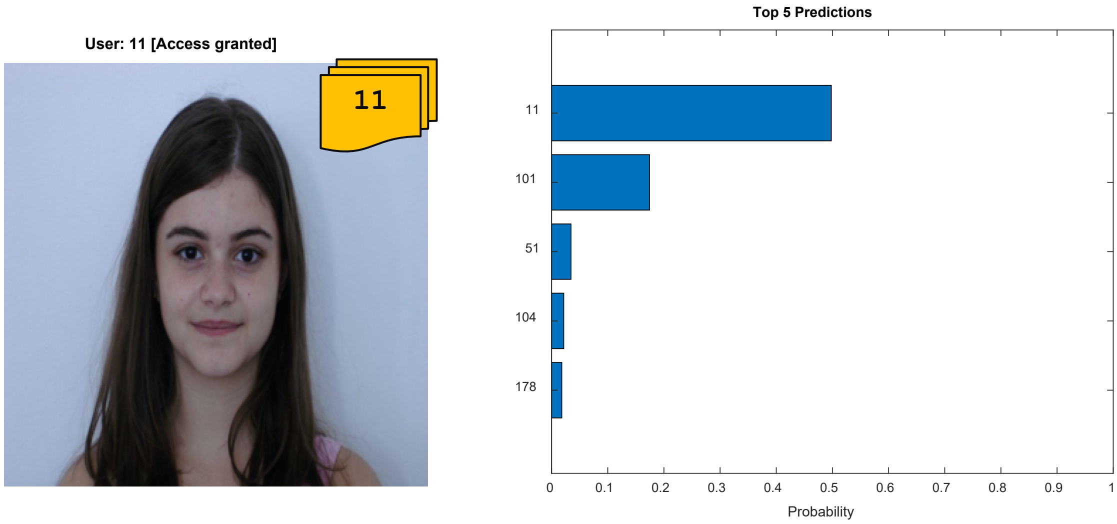


Figure 4.15. Correct prediction + Authorized example 1

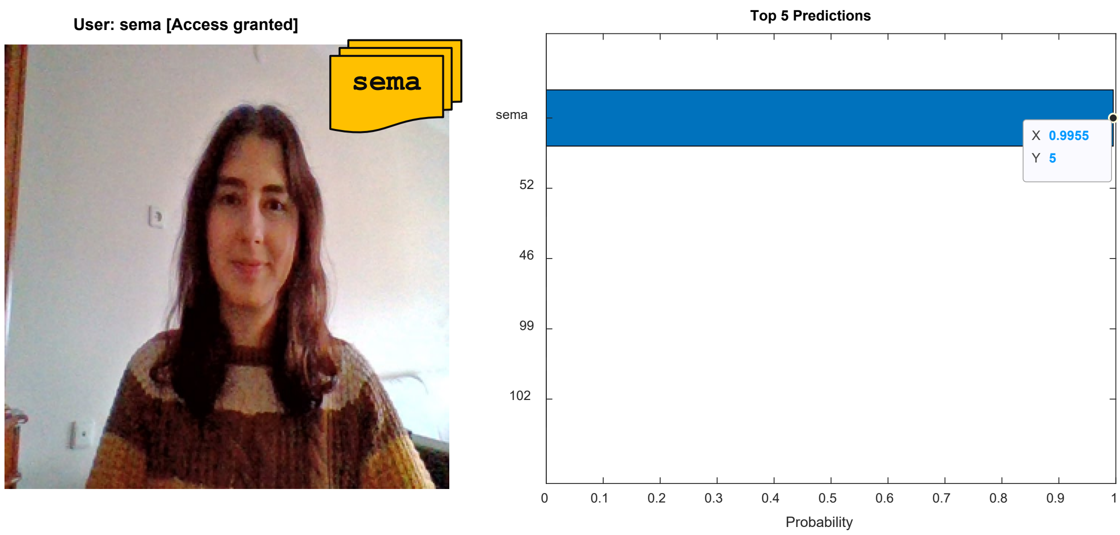


Figure 4.16. Correct prediction + Authorized example 2

The result implies that, the system is able to identify who is presented. An important remark, here, is that, the presented image has never used before in training. The result also verifies the k-fold accuracies which were obtained before. The Figure 15 and 16 shows the individuals who we have given authorization.

The second test is done for individuals which are not authorized to use the vehicle. In Figure X, the face recognition, again, correctly recognize who that individual is, but since the user '50', not authorized, it is displayed as imposter.

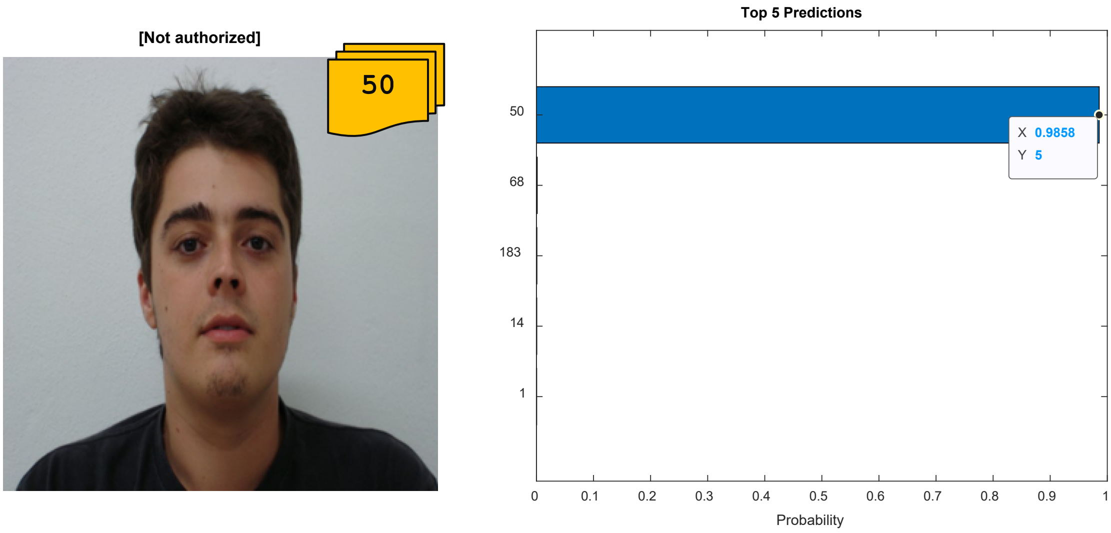


Figure 4.17. Correct prediction + Not Authorized example

In practical usage, the imposters will be mostly the images that had never encountered before. Here, the prediction percentage plays a crucial role. In Figure 18 below, the image which never used in training, is presented to the system. Although, the presented image's prediction percentage indicates it is '107', its prediction strength is approximately 3%, which extremely low. Obviously, having top similarity does not mean presented image is genuine and here, the percentage threshold enables to reject the imposter user. In short, In Figure 18, below, given input does not have enough prediction strength to be classified an individual in trained dataset (which did not exist in trained dataset), and not authorized as well.

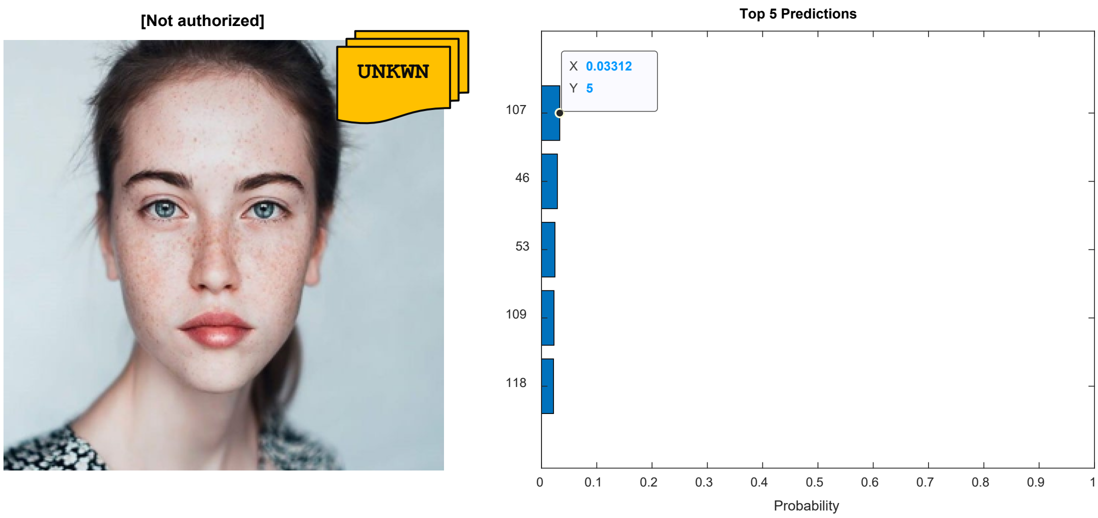


Figure 4.18. No prediction + Not Authorized example

4.2 Drowsiness Detection

As discussed in Part 1: Introduction, driver's drowsiness can be easily detected by monitoring the eye status which is either 'open' or 'closed'. To accomplish that, the eyes must be located. Then, after trying various methods, we decided to use similarity matching with a given open eye template, as explained in the end of Section 4.2.4.

4.2.1 Algorithms for Eye Detection

Similar to detecting human faces, eye pairs can be detected using Viola-Jones algorithm, and this is implemented with **`Computer Vision Toolbox`** in **`MATLAB`**. Here, to detect the eye pair, **`vision.CascadeObjectDetector`** is called with '**`EyePairBig`**' parameter. The object detector uses Viola-Jones algorithm. The eye pair can be detection in all possible states, as shown in Figure 4.19 below.

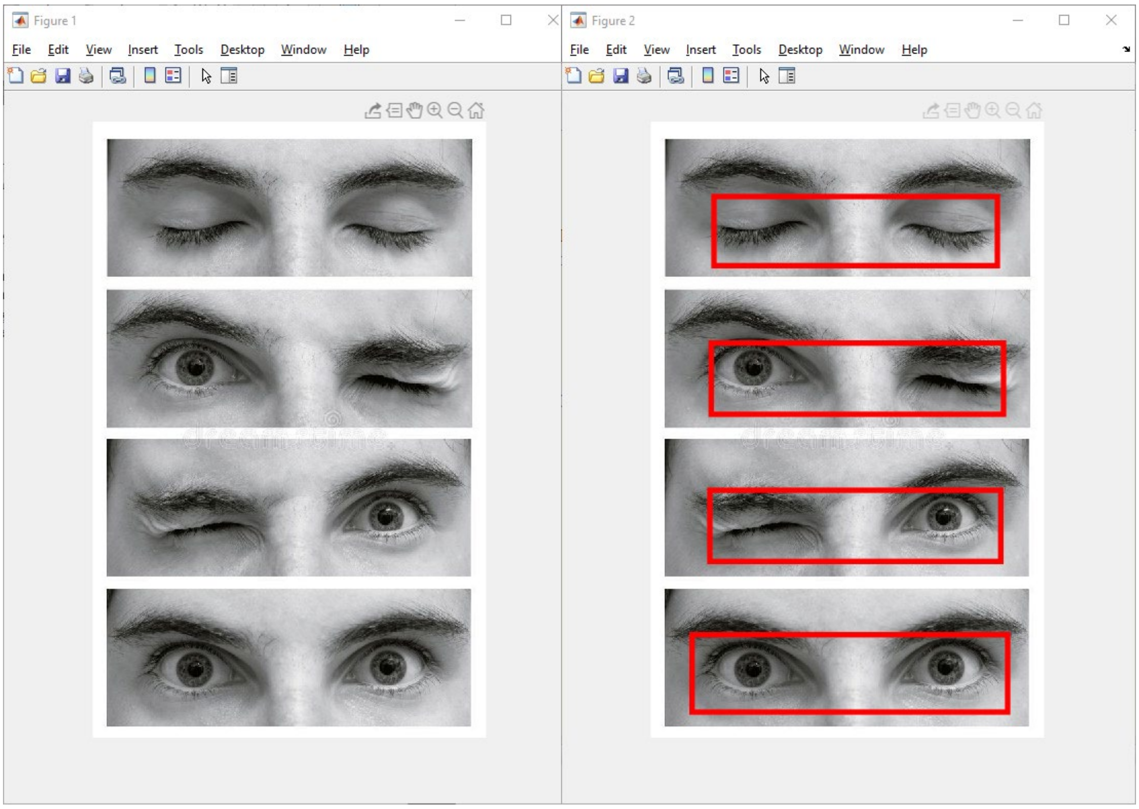


Figure 4.19. Eye-pair detection of given image

4.2.2 Frame Acquisition and Eye Tracking Algorithm

The state of the eye pair of the driver is monitored with video stream. Again, the webcam is used. The frames are acquired using **`Image Acquisition Toolbox`** in **`MATLAB`**. In the frames, the single image is captured for each given time interval, and the eyes are detected within the captured image.

We created a motion-based system for detecting and tracking multiple moving object that is eye pair for our project. The steps of multiple object tracking can be divided into two parts: Detecting moving objects in each frame and tracking the moving objects from frame to frame.

The detection of moving objects uses a **`Kanade Lucas Tomasi`** algorithm. We create a **`vision.PointTracker`** object. It is tracking points in video using **`Kanade-Lucas-Tomasi (KLT)`** algorithm which is a feature- tracking algorithm. This object works particularly well for tracking objects that do not change shape and for those that exhibit visual texture. Figure 4.20 illustrates the flowchart of KLT algorithm.

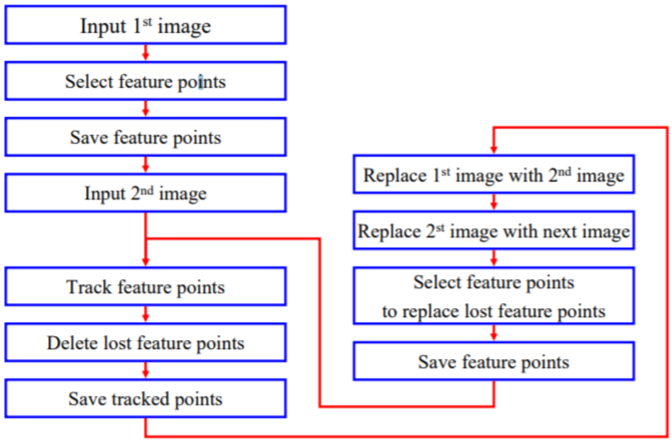


Figure 4.20. Flowchart of **`Kanade Lucas Tomasi`**

To eliminate points that could not be reliably tracked by **`PointerTraker`**, it has efficient property which is **`MaxBidirectionalError`**. It is specified with as a scalar. In MATLAB, it is recommended to use the value between 0 to 3. Setting the value to less than **`inf`**, it provides the tracker tracks each point from the previous to the current frame. It then, tracks the same points back to the previous frame. The object calculates the bidirectional error. This value is the distance in pixels from the original location of the points to the final location after the backward tracking. The corresponding points are considered invalid when the error is greater than the value set for this property. In project, we considered selecting the value of 3 pixels.

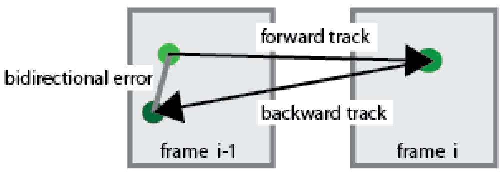


Figure 4.21. Point tracker bidirectional error illustration

Once tracking is desired to be completed, the system resources such as memory, file handles, or hardware connections can be released by **`release(obj)`** function call.

4.2.3 Processing Methods

→ *Using mean distribution of each eye*

Figure 4.22. Open Eye and Closed Eye (single)

First, we have tried with measuring mean distribution of intensity values of pixels for images which contains open and closed eyes, for each single eye. The single eyes are cropped from taken image and the mean distribution is calculated. When the distribution is plotted for each state of eye, we obtain the plot in Figure 4.23. As seen in the plot, the characteristics of each state is different and can be distinguished. There is a significant distance between open eye (shown with red in Figure 4.23) and closed eye (shown with blue). In the Figure 4.23, the interval [0, 60] corresponds to the vertical location of eye.

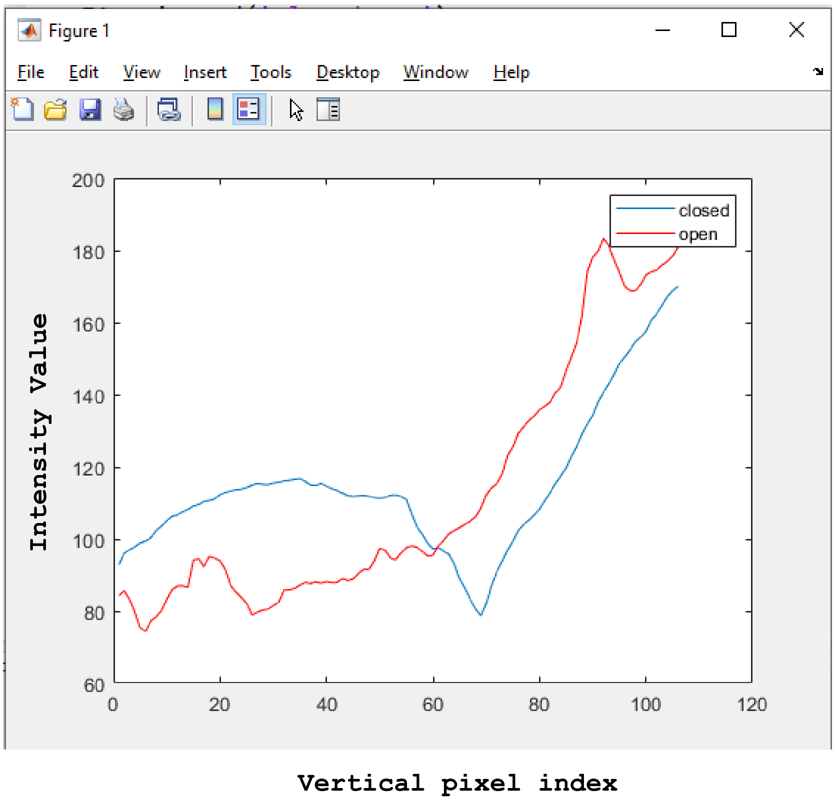


Figure 4.23. Mean plot of Open and Closed eye

→ *Using mean distribution of eye-pair*

Then, we have measured mean distribution of intensity values for images which have eye-pairs, with open and closed states.

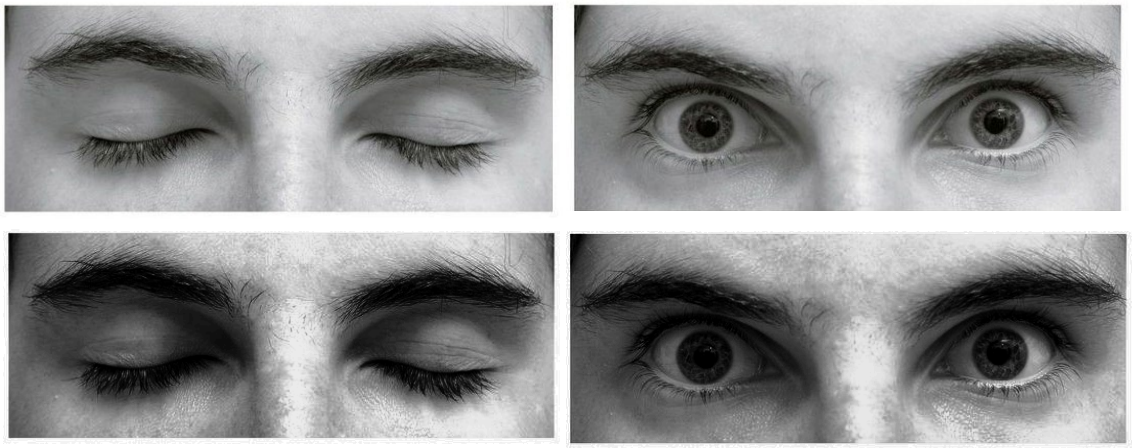


Figure 4.24. Open and Closed eye-pairs and their histogram equalized versions

Figure 4.24 below shows the mean distribution of images with open and closed eye pair.

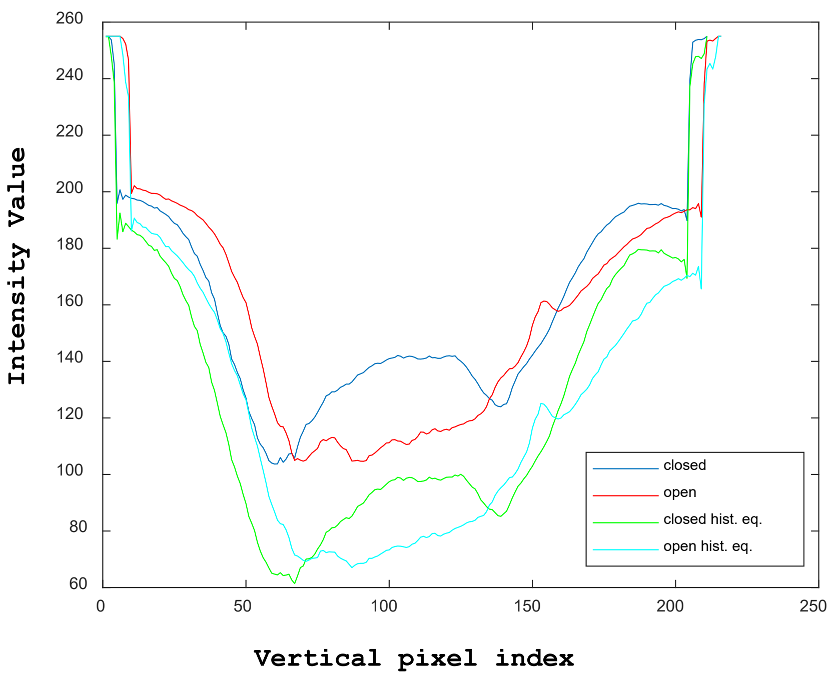


Figure 4.24. Open and Closed eye-pairs

→ *Working with both eyes (comparison)*

Here, we analyze the behavior of the curve of mean distribution for each possible state of eye-pair. In the figure 25.b, we see that the area under the curve is larger when both eyes are closed, compared to other situations. On the other hand, having both eyes open produces a plot with lower area under the curve, as seen in Figure 25.b

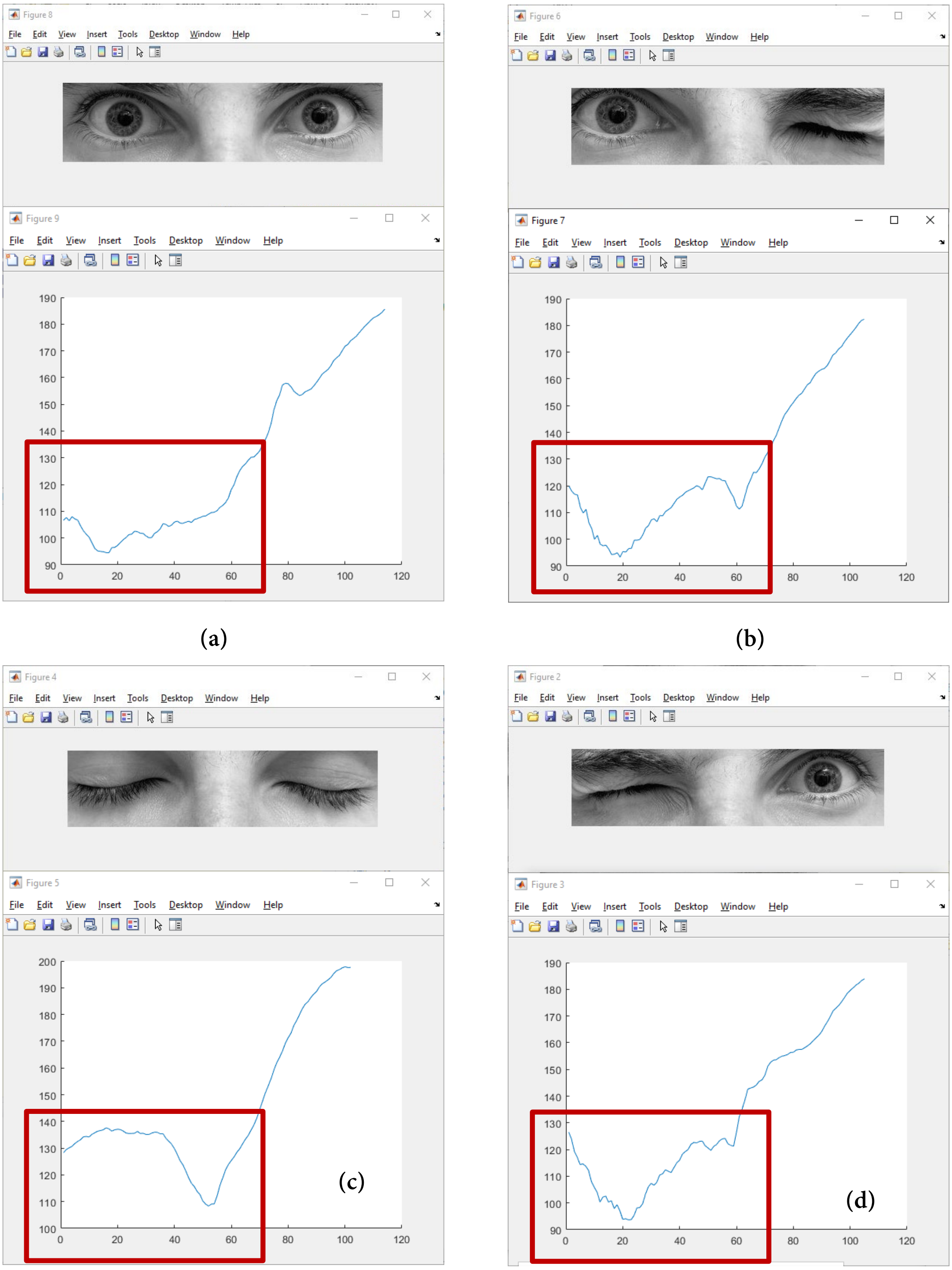


Figure 4.25. Determining state of eye based on behavior of curves

→ *Working with Normalized Histograms*

Here, the cropped single eye images are normalized and converted into the range of [0, 1]. Figure 4.26 shows the single eyes with open and closed states.

Figure 4.26. Open and Closed eye

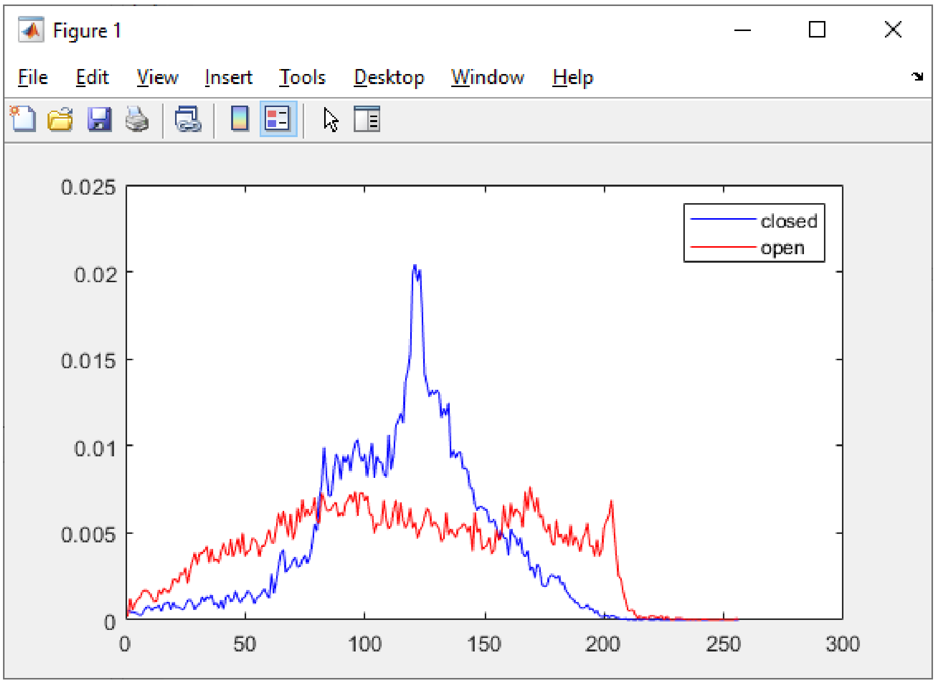


Figure 4.27. Plot of normalized Histograms of Closed and Open Eye

However, when it is used for both eyes at the same time, it is not as distinguishable as it was in single eyes.

Figure 4.28. Both eyes are used

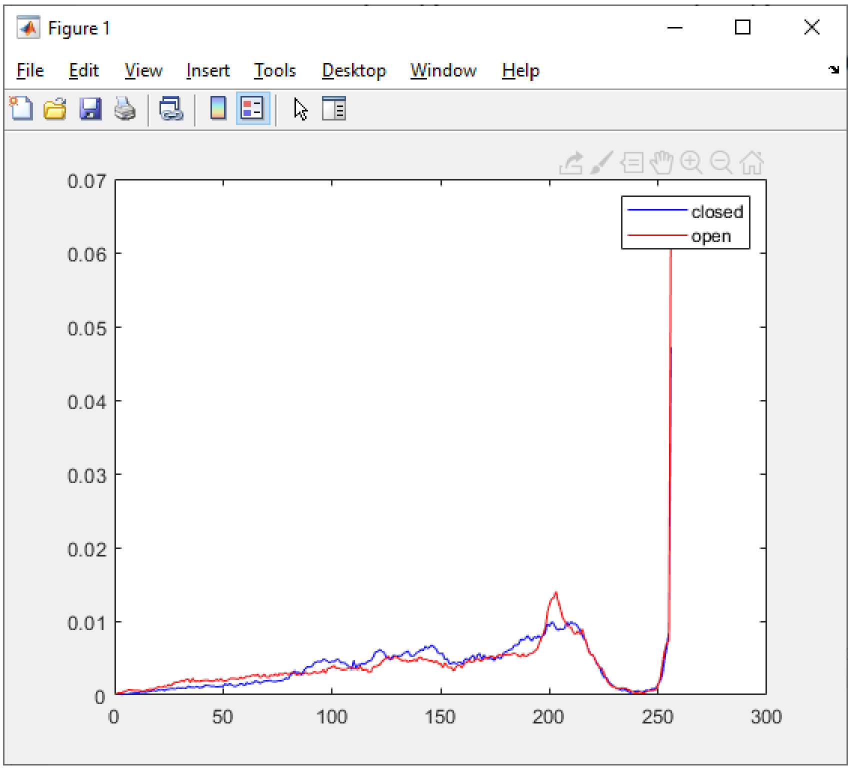


Figure 4.29. Plot of both eyes with normalized histograms (not distinguishable)

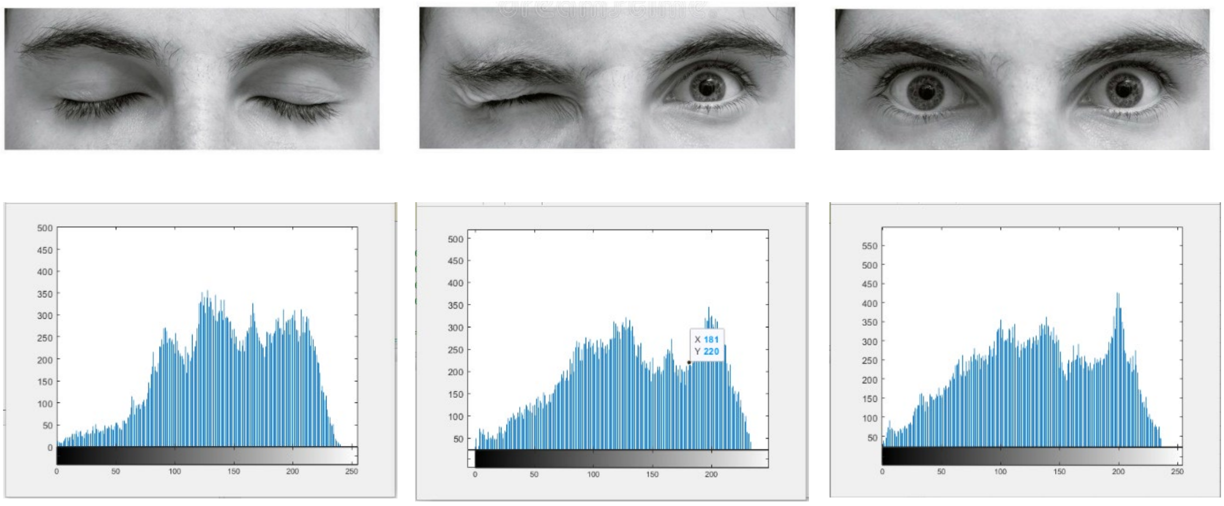


Figure 4.30. Normalized histograms of various states (not distinguishable)

→ *Working with LBP*

As we have learned from literature, the LBP could be used to determine the state of eyes. Thus, we implemented LBP for different states (both open, both closed, only right eye open and only left eye open). With both eyes open (state 4) and both eyes closed (state 2) the Squared Error of LBP Histograms are in Figure 32.

Figure 4.31. Open and closed eye pairs

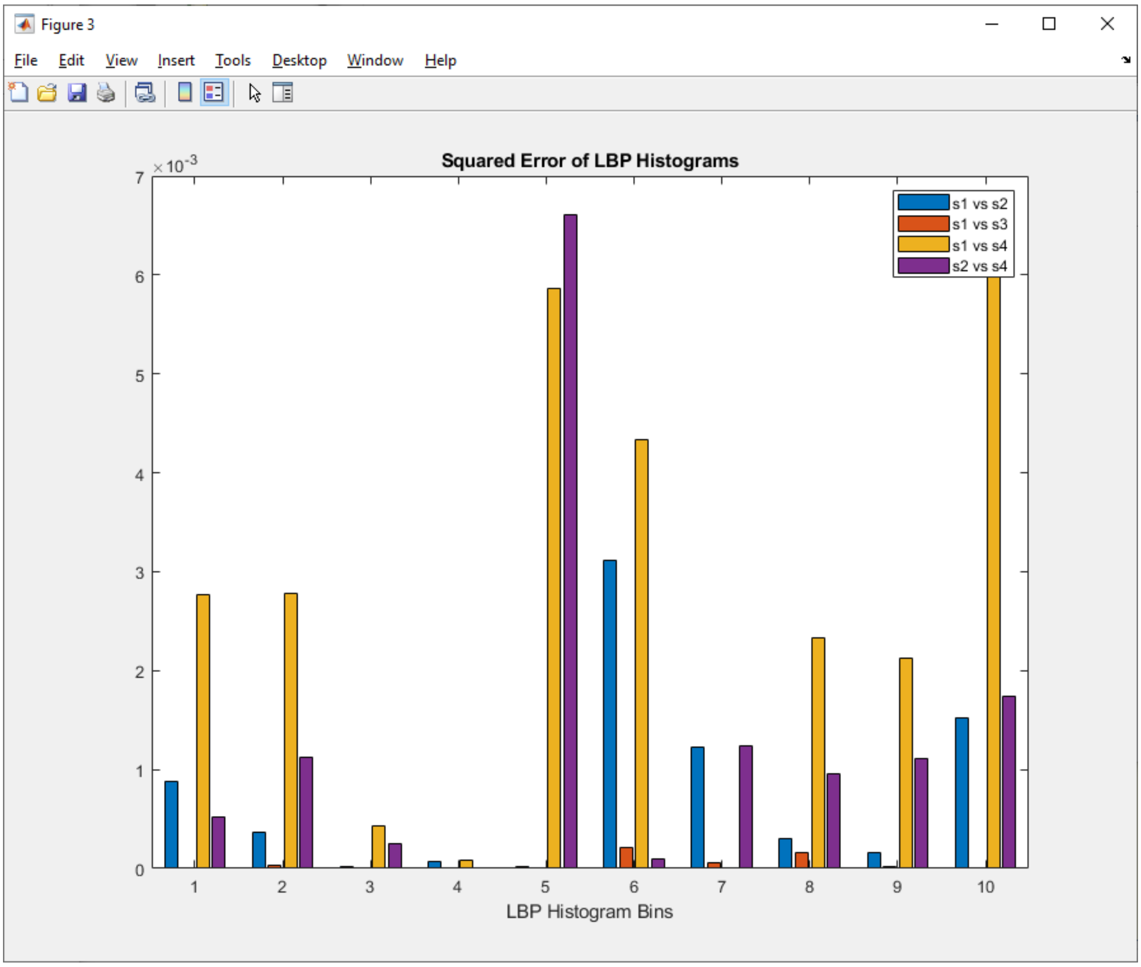


Figure 4.32. Squared Error of LBP Histograms (Comparisons between various states)

→ *Similarity Matching with given Template*

Another way of deciding whether the eye is closed or not, can be done with comparing similarity of given eye pair with open eye pair. Here, the open eye template image can be obtained as shown in Figures 4.33.a – 4.33.e with given processes as described in figures.

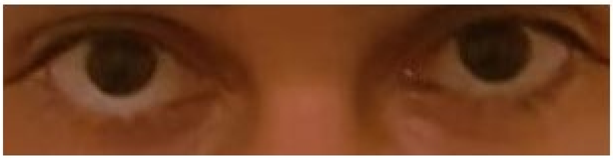


Figure 4.33.a– Cropped image from drive's face

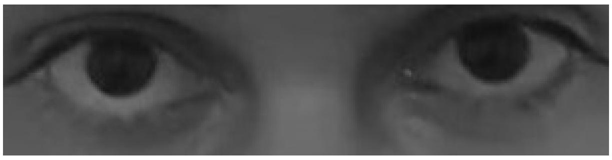


Figure 4.33.b – Converted into grayscale image

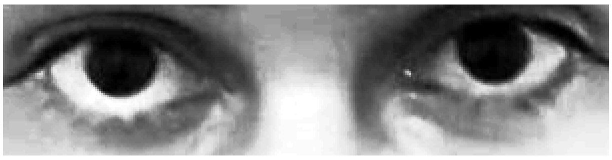


Figure 4.33.c – Histogram equalized image

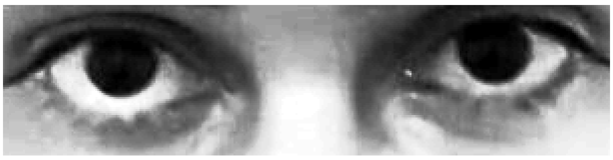


Figure 4.33.d – Blurred image

Figure 4.33.e – Binarized image (i.e. black and white)

Figure 4.33. – Image processing of driver's eye-pair; (a) Eyes are located and cropped; (b) Cropped image is converted into gray scale to have color independent and better performance; (c) Image is enhanced with histogram equalization; (d) Enhanced image is blurred to get rid of noises; (e) Blurred image is converted into binarized image with specified threshold.

Once the template image is obtained, it will be used for comparison with frame images constantly. In this process, it is required to keep the eye-pair open when taken the template image, for system to work properly.

4.2.4 Evaluation Metrics

We have decided to use *similarity matching with given template*. The open eye pair template is loaded by program, as it will be used for comparisons. As mentioned in section 4.2.2, the frame is obtained via laptop computer webcam, in live stream. The program detects the eye pair and keep tracking it until the program terminates.

The similarity is calculated by two-dimensional (2-D) correlation coefficients r which is shown in Equation X, below:

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \overline{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \overline{B})^2\right)}}$$

Equation 4.1 – Correlation coefficient

where A and B are **`image_captured`** and **`image_template`**, respectively and $\bar{A}$ and $\bar{B}$ are the the mean or average of matrix elements which are pixel values of images A and B, whose dimensions are m and n.

The program takes snapshots of live video stream in each frame. For every 5 frames, we check the status of the eye pair by comparing it with the template. For our notebook computer, 5 frame corresponds to approximately three seconds. If the images taken in these consecutive fives frames are classified as closed eye, then the program alerts.

For our tests, we observe that, the closed eye pair is distinguished when correlation value ***`r`*** is lower than `0.10`. Hence, we specified threshold value of ***`r`*** to `0.10` and the program detects the closed eye when the correlation value ***`r`*** is lower than specified value.

Figure 4.34. – Correlation score is 0.0453

Since the similarity can differ across different individuals as well as different light conditions, the template is keep updated when ***`r`*** rises above `0.50`. This means that, the update happens only when it is made sure that the eyes are open. Hence, template updating allows us to obtain more robust performance. Otherwise, the lower similarity would have obtained even though eye pairs were open state.

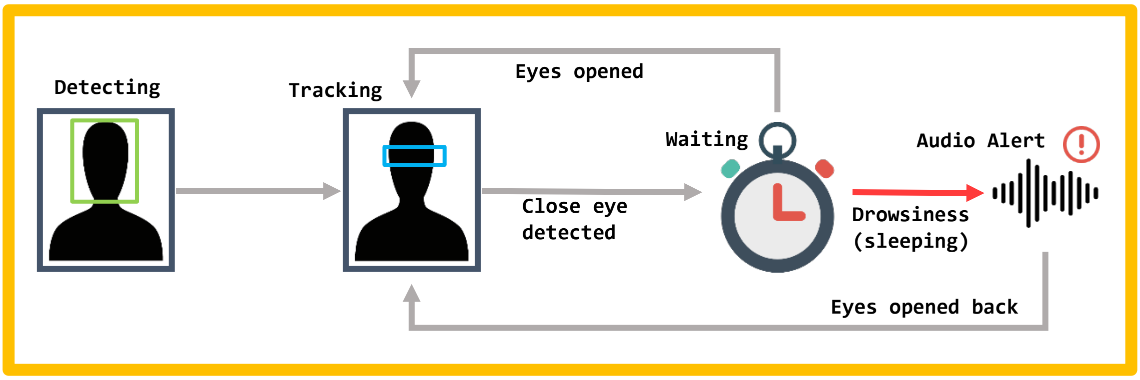


Figure 4.34 – Alert mechanism

To provide better understanding, in real-life situations, prior condition is detection of eye pair being close, in longer duration than specified. If the duration is exceeded, it will trigger the alert mechanism.

If the alert condition is satisfied, the system the activates warning state which it bleeps as long as the eyes are kept closed. The Figure 34, above, illustrates the mechanism.

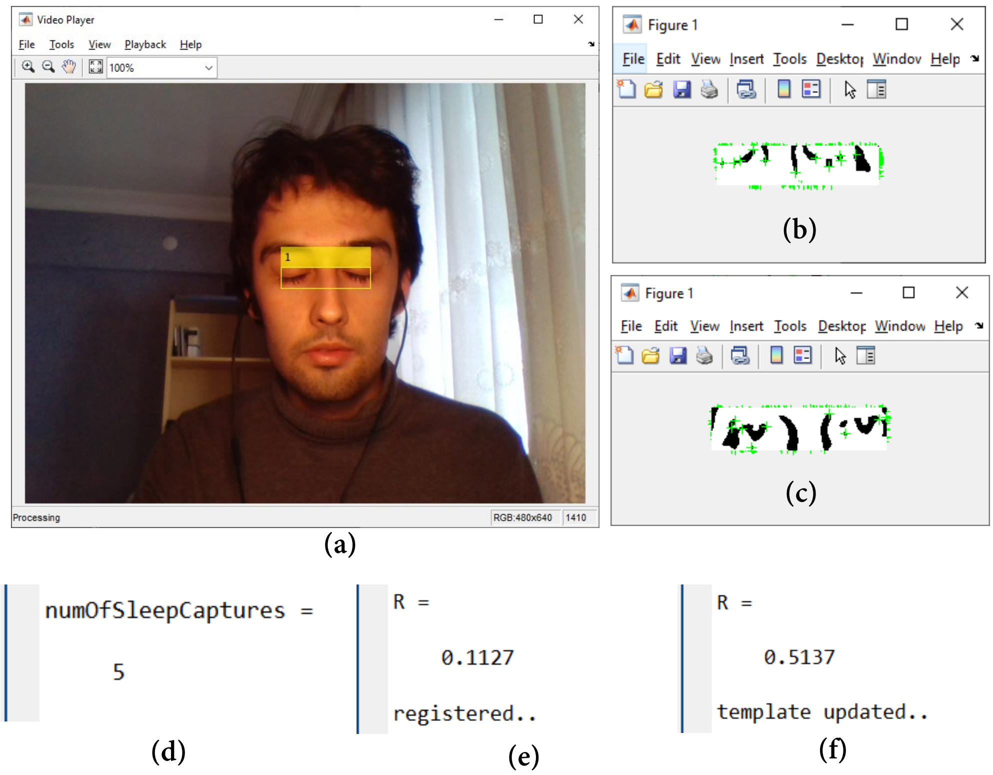


Figure 4.35 – Drowsiness detection demo

(a) Sleeping driver whose eye-pair are detected;

(b) closed eye pair template; **(c)** open eye pair template;

(d) number frames which eye pair is closed;

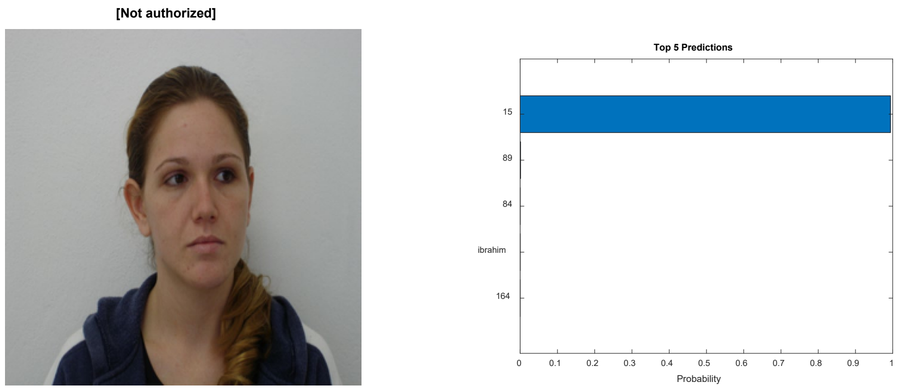
(e) Driver's template is registered for the first time;

(f) Template is updated when eye pair is open and R > 0.50

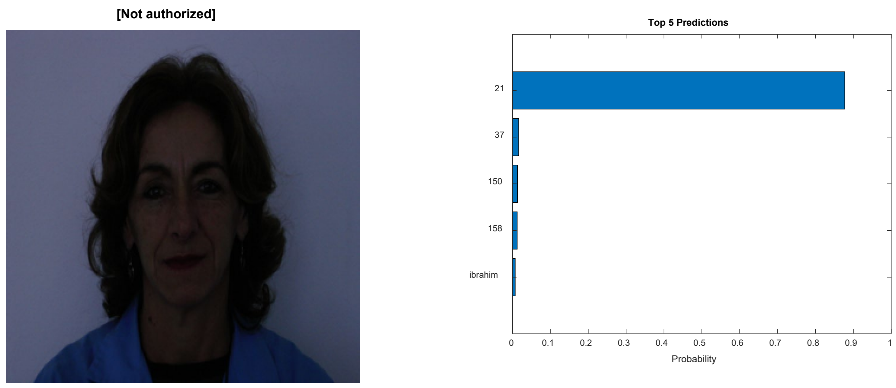
5. RESULTS AND EVALUTIONS

The inceptionv3 architecture was trained in MATLAB R2019a on laptop computer having Intel i7 4720 HQ @ 3.60 GHz CPU and Nvidia GTX 970M GPU with operation system of Windows 10 SL. With selected parameters, the training takes about 7 hours. To obtain quicker but less accurate models, having learning rate of five times larger with less epoch can be selected (Table 4.1), or smaller dataset size (e.g. EFI part 1) can be used.

The facial recognition validation results are presented below. For the test images we have done excluded images of trained database, we obtained great accuracies in recognition percentage confidence. What we have observed is that, when the illuminance level is closer to training images, the performance was even better, as seen in Figure 5.1.a. However, when having lower illuminance level, the system was still accurately predicting the subject.



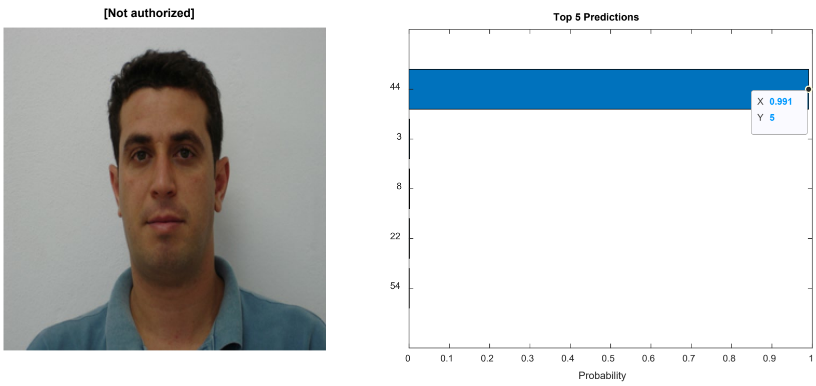
(a)



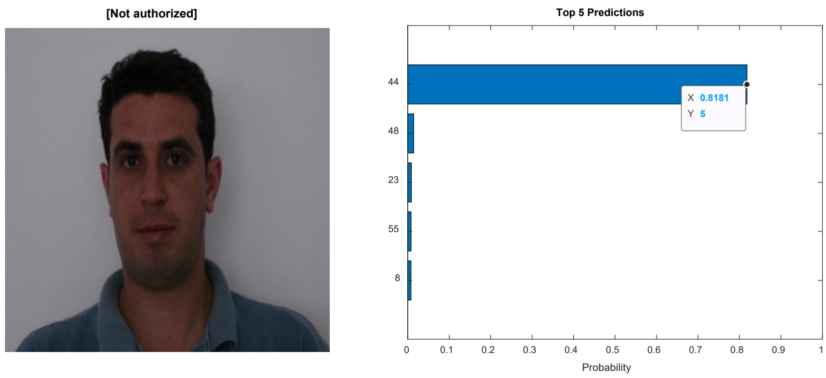
(b)

Figure 5.1 – Predictions with very high accuracies;
(a) Illuminance level similar to illuminance level of trained image;
(b) Illuminance level is lower compared to trained image

For the same prediction percentage for the same individual in different illuminance level is presented in Figure 5.2



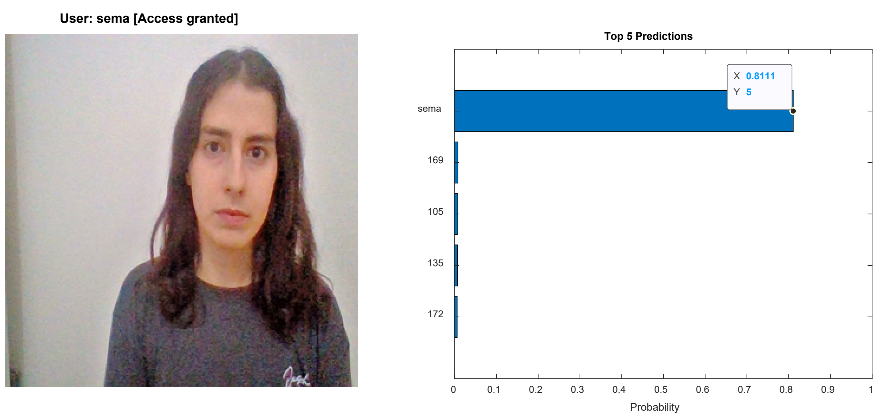
(a)



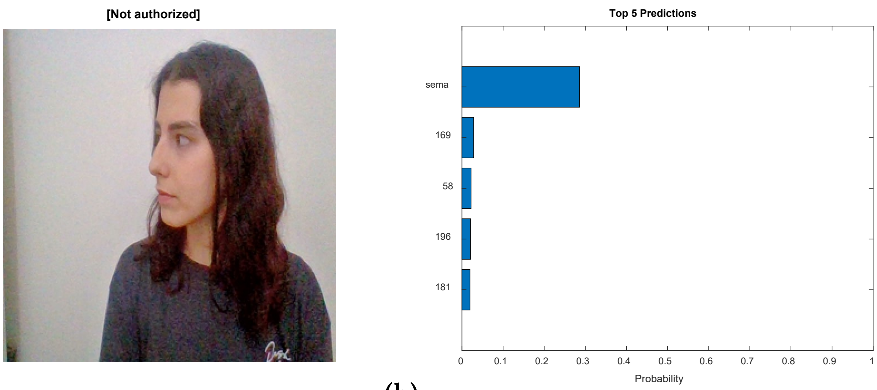
(b)

Figure 5.2 – Subject 44 in **(a)** normal illuminance level; **(b)** low illuminance level

The other test is done authorized person (*Sema)* in different perspective. The top match is correctly determined; however, the confidence level was lower than specified threshold. For security purposes, the threshold was not kept too low. The subject needs to have frontal view to the camera with specified distance to obtain higher match score, consequently having access granted.



(a)



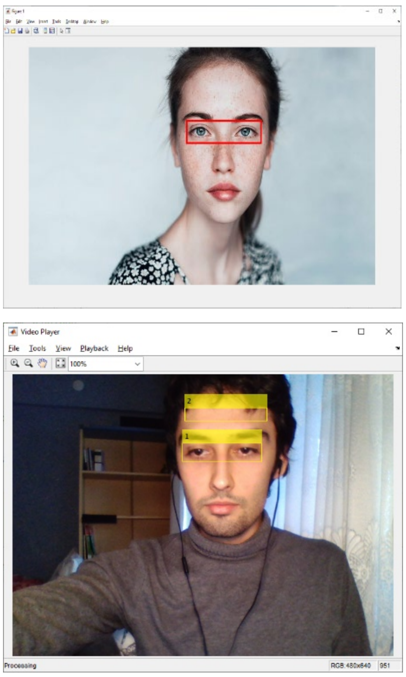
(b)

Figure 5.3 – Subject *Sema* in **(a)** frontal view; **(b)** side view

For eye detection, the algorithm can be improved with having better acquisition methods. With provided images and real-time input stream, it was working fine under not nodding, or keeping the head stable state. Keeping the head stable, the problem was able to detect the eye-pair successfully, both in given image and real-time input stream.

Moving the head very quickly give rises an exception (error) for program, because the hardware (i.e. camera, CPU/GPU power, frame rate etc.) were not capable of detecting eye in quick movement of head.

Another constraint for drowsiness detection is that the lightning of environment and direction of lightning. We assume the light is reflected in frontal direction. Also, the driver needs to position in 35-50 cm distance to the camera (at least, for our machine), which is an appropriate distance between driver and wheel, considering the vehicle application.



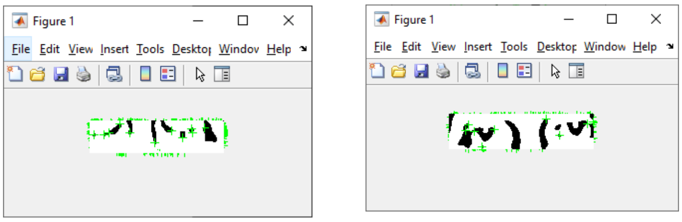


Figure 5.4 – template images in real-time input

As a result, we have observed that, the system was able to produce a sound alarm when it senses the eye pair is closed. Here, we specified the allowable closed eye duration to be maximum 3 seconds, which is reasonable duration, because having longer than that, will increase the likelihood of accident dramatically. The program keeps monitoring the eye pair as long as it detects an eye pair, or it is terminated.

Figure 5.5 – Correct detection & incorrect detection

6. CONCLUSION AND FUTURE WORK

In this project, the applications of image processing and machine learning, specifically convolutional neural networks (CNN), techniques on vehicle crimes and accidents due to drowsiness are aimed. Use of biometric identification methods provides a high-level degree of security. Among these identification methods, one the most popularly used is facial recognition. With facial authentication system, the user uses his face to verify his identity and its applications are widening each day.

As a remedy to increasing vehicle crimes, we have designed a facial recognition system which leverages from CNN. With this system, user genuine drivers can be authenticated. We have decided to select inceptionv3 pretrained CNN architecture because it provided a sweet spot performance on computational requirement vs accuracy. We have extended the selected datasets FEI Face Database, with our images. Then the architecture is trained with 200 + 2 individual dataset. The parameter of architecture was selected based on the obtained accuracies from plentiful of experiments. Then, we have conducted several experiments to see the validity of accuracy from k-fold training process, which was approximately 99%. To test the validity with practical usage, we provided excluded images from database and provided them to the trained architecture. The trained model was supremely successful in recognition task of individuals which the dataset used included. For our purpose, we desired to distinguish a person genuine or imposter. Accordingly, we specified few users, including our names, as authorized user. Then, the program successfully displayed whether a given image is authorized or not. For our tests, we have used laptop webcam when registering new user to the system. A far better performance could be obtained using a hardware that is specifically designed for this purpose. With such additional hardware, better images could be acquired, and that will boost the performance of the system even more. In facial recognition, another future contribution would be omitting the background of taken facial image using Artificial intelligence, as having background not solid have some impact on the overall potential.

To have impact on drowsiness which is one of the most serious reason that leads to deadly accidents, we have endeavored to extend our system to be used in drowsiness detection. One of the simplest ways of measuring drowsiness, i.e. sleeping state, is classifying the state of eyes being open or closed. Accordingly, we have attempted various techniques to classify the eye states. We decided to use matching similarity between given eye pair with opened-eye template image was the most appropriate choice. The system keeps monitor the eye pair in real time video stream, and when the closed eye pair is detected, it produces warning audio state. There is plenty room for improvement in drowsiness mechanism. The mechanism is still affected by noise, frame drops, or distance to the camera. The processing certainly can be improved using additional image processing techniques. Again, we planned to obtain the video stream with external camera with higher quality if our proposal was accepted by Tubitak.

With our efforts, it is hoped that image processing and machine learning techniques would improve the safety of both vehicles from stealing, and people from drowsiness related accidents involving injuries or deaths.

7. References

Abtahi, S., Hariri, B., & Shirmohammadi, S. (2011, May). Driver drowsiness monitoring based on yawning detection. *In 2011 IEEE International Instrumentation and Measurement Technology Conference* (pp. 1-4). IEEE.

Akalin, V. (2003). Face recognition using Eigenfaces and neural networks. The Middle East Technical University.

AlMarri, S. B. S. (2019). Real-Time Facial Emotion Recognition Using Fast R-CNN.
Baharu, b. (2013). Driver drowsiness detection by using webcam.

Bashbaghi, S. (2017). Face recognition in video surveillance from a single reference sample through domain adaptation (Doctoral dissertation, École de technologie supérieure).

Björgvinsdottir, H. B., & Seibold, R. (2016). Face Recognition Based on Embedded Systems. Master's Theses in Mathematical Sciences.

Dawaga, M. E. (2016). Automatic detection of eye blinking using the generalized ising model.
Delbiaggio, N. (2017). A comparison of facial recognition's algorithms.

Ghahramani, F. (2015). Face Recognition: An Engineering Approach.

Hu, J. (2017). A personal facial expression monitoring system using deep learning (Doctoral dissertation).

Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., & Jiang, S. (2018). Real-time driver drowsiness detection for android application using deep neural networks techniques. Procedia computer science, 130, 400-407.

Karaaba, M. F. (2016). Face recognition in low-resolution images under small sample conditions with face-part detection and alignment (Doctoral dissertation, University of Groningen).

Lawoyin, S. (2014). Novel technologies for the detection and mitigation of drowsy driving.

Le, K. D. (2019). A study of face embedding in face recognition.

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5325-5334).

Liu, Q., & Wang, J. Facial Expression Recognition based on CNN.
Nabatchian, A. (2011). Human face recognition.

Omar, S. H. A. (2016). BIOMETRIC SYSTEM BASED ON FACE RECOGNITION SYSTEM.

Pavani, S. K. (2010). Methods for face detection and adaptive face recognition. Universitat Pompeu Fabra.

Reddy, B., Kim, Y. H., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 121-128).

Rodríguez Navarro, D. (2017). Multimodal Deep Learning methods for person annotation in video sequences (Bachelor's thesis, Universitat Politècnica de Catalunya).

Shepley, A. J. (2019). Deep Learning For Face Recognition: A Critical Analysis. arXiv preprint arXiv:1907.12739.

Subudhi, K. K. K., & Mishra, R. (2011). Human Face Detection And Recognition. Department of Electronics and Communication Engineering National Institute of Technology, Rourkela.

Subudhi, K. K. K., & Mishra, R. (2011). Human Face Detection And Recognition. Department of Electronics and Communication Engineering National Institute of Technology, Rourkela.

Tadesse, E. F. (2013). Drowsiness Detection for Driver Assistance (Doctoral dissertation, Oklahoma State University).

Takalahti, A. Learning to Detect Drowsiness in Drivers.

Tunçer, Ö., Güvenç, L., Coşkun, F., & Karsligil, E. (2010, October). Vision based lane keeping assistance control triggered by a driver inattention monitor. In 2010 IEEE International Conference on Systems, Man and Cybernetics (pp. 289-297). IEEE.

Türkan, M. (2006). Human face detection and eye location in video using wavelets (Doctoral dissertation, bilkent university).

Valero, H. G. (2016). Automatic Facial Expression Recognition. School of Computer Science, The University of Manchester, Master Thesis.

Vural, E. (2009). Video based detection of driver fatigue (Doctoral dissertation).