An Approach Towards Real-time Evaluation of Eye Blink and Gaze Direction using SVM, Viola-Jones Algorithm and Hough Transform

Abdullah Al Noman

B.Sc. Engineering in Computer Science & Engineering

Project (Course Code: CSE 800)

Submitted to the Department of Computer Science & Engineering, University of Chittagong in partial fulfillment of the requirements for the degree of B.Sc. Engineering in Computer Science & Engineering.

Department of Computer Science & Engineering

University of Chittagong

February, 2019.

An Approach Towards Real-time Evaluation of Eye Blink and Gaze Direction using SVM, Viola-Jones Algorithm and Hough Transform

Abdullah Al Noman

B.Sc. Engineering in Computer Science & Engineering

Project (Course Code: CSE 800)

Department of Computer Science & Engineering,

University of Chittagong.

CODE:	CODE: STUDENT ID: 15701016 Name: Abdullah Al Noman Session: 2014-15 Hall: Shahid Abdur Rab Hall Date of Submission: 03/03/2019 Signature of Student:

CANDIDATE'S DECLARATION

I hereby declare that this project report has been composed by me and has not been submitted, in whole or in part, in any previous application for a degree, except where it states otherwise by references or acknowledgement.

Abdullah Al Noman

(Candidate)

APPROVAL

This project report entitled “An Approach Towards Real-time Evaluation of Eye Blink and Gaze Direction using SVM, Viola-Jones Algorithm and Hough Transform" prepared and submitted by Abdullah Al Noman, ID-15701016 in partial fulfillment of the requirements for the degree of B.Sc. Engineering in the Department of Computer Science and Engineering, University of Chittagong has been examined and is recommended for approval and acceptance.

Supervisor

Muhammad Anwarul Azim
Professor,
Department of Computer Science and Engineering,
University of Chittagong.

TABLE OF CONTENTS

Contents **Page No.**

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my supervisor Muhammad Anwarul Azim, Professor, Department of Computer Science & Engineering, University of Chittagong, for his consultation & excellent guidance throughout the project. Without his encouragement and support this project wouldn't have been possible. His guidance helped me towards completing my work.

I have to appreciate the support given by my family members and classmates. Thanks to their comments and advices.

ABSTRACT

A real-time approach to evaluate eye blinks and eye gaze direction in a video stream from a standard camera is proposed. Classification based eye blink detection has been done in this project. Histogram of Oriented Gradient (HOG) is used as a feature descriptor. Eigen face features are detected in the Region of Interest (ROI). From rectified face eye pair are detected. Support vector machine (SVM) is used as classifier to classify closed eyes as positive class and opened eyes as negative class as they are binary classes. With the help of the HAAR feature selection method in Viola-Jones object detection algorithm, Adaboost training has been implemented to integrate images to detect objects such as one single eye. Hough transform edge detection has been implemented to detect the edges of the pupil in a single eye and draw a circle around the pupil. As detecting the pupil, metric distances from the eye-edges were measured to evaluate gaze direction.

CHAPTER 1

INTRODUCTION

1.1 Synopsis

Evaluation of eye blinking is important for instance in systems that monitor a human operator vigilance, e.g. driver drowsiness, in systems that warn a computer user staring at the screen without blinking for a long time to prevent the dry eye and the computer vision syndromes, in human-computer interfaces that ease communication for disabled people, or for anti-spoofing protection in face recognition systems. [1]

There's been many methods for detect eye blinks from video stream. To estimate eye blinks, per-frame sequence of eye openness is measured. Considering closed eyes as positive class and opened eyes as negative class, support vector machine classifies through the descripted features by histogram of oriented gradient. [2]

Human visual attention mostly depends on eye gaze direction. [3] If we can measure the gaze direction, then we can approach towards evaluating human concentration or attention on certain points.

1.2 Motivation

Automated eye blink detection through digital image processing needs classification techniques to classify eye openness. We used binary classification for evaluating open and close eyes. Human eye blink differs from person to person in terms of speed of closing and opening, degree of squeezing, duration of blink etc.

There are two main parameters of blinking: frequency and duration. Average frequency of blinking of an adult is 15-20 blinks/min but there are only 2-4 blinks/min physiologically needed. Each eye blink lasts approximately 100-400 ms. [4] It's difficult to measure the time duration between opening and closing of eyes. A person's visual attention can be arbitrarily evaluated through eye gaze direction.

Gaze region estimation along with head posture will be more precise to estimate human behavior. Along with facial landmarks eye blink, eye gaze can classify human expressions. Gaze-based attention system for multi-human multi-robot interaction can be done. Semi-automated vehicle intelligent system can be generated. Smart braking system along with smart auto-pilot system can be introduced.

Both eye blink and gaze direction can be used to evaluate human behavior. Applications such as human concentration evaluation, driver gaze direction, vehicle automation, children attention measurement etc. In this project HOG, SVM, Viola-Jones algorithm, Hough transform has been used for feature extraction and classification.

1.3 Background of the Project

Digital image processing has been a great influence over computer vision. In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images. [5] As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing.

Many of the techniques of digital image processing, or digital picture processing as it often was called, were developed in the 1960s at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland, and a few other research facilities, with application to satellite imagery, wire-photo standards conversion, medical imaging, videophone, character recognition, and photograph enhancement. The cost of processing was fairly high, however, with the computing equipment of that era.

That changed in the 1970s, when digital image processing proliferated as cheaper computers and dedicated hardware became available. Images then could be processed in real time, for some dedicated problems such as television standards conversion. As general-purpose computers became faster, they started to take over the role of dedicated hardware for all but the most specialized and computer-intensive operations. With the fast computers and signal processors available in the 2000s, digital image processing has become the most common form of image processing and generally, is used because it is not only the most versatile method, but also the cheapest.[6]

Evaluating human attention has to be upon various aspects. Eye blink, eye gaze direction, head posture, gaze region etc. can be the key aspects of this system. Correlation between head and eye movement have shown inter-person variation in the degree to which the head serves as a proxy for gaze. Extracting eye blink information is a tricky job. Digital image processing algorithms and classification techniques made them come true.

1.4 Problem Domain

Naturalistic driving studies have shown that a driver's allocation of visual attention away from the road is a critical indicator of accident risk. [7] Such work would suggest that a real-time estimation of driver's gaze could be coupled with an alerting system to enhance safety when the driver is overly distracted or inattentive.

From an image processing perspective alone, difficulties involve the unpredictability of the environment, presence of sunglasses occluding the eye, rapid changes in ambient lighting including situations of extreme glare resulting from reflection, partial occlusion of the pupil due to squinting, vehicle vibration, image blur, poor video resolution etc. [8]

As the level of vehicle automation continues to increase through Advanced Driver Assistance Systems as well as other higher forms of automation, freeing available resources from the primary operational task, drivers are expected to be increasingly allowed to glance away from the roadway for greater periods. When the need arises to orient the driver to the roadway, different alerting strategies may be advantageous. Such work would suggest that a real-time estimation of driver's gaze could be coupled with an alerting system to enhance safety. [9]

Furthermore, preschool children concentration evaluation is a difficult task. Children growth specially depend on their childhood monitoring. Later life which is the outcome of childhood. So better monitoring will give better childhood of a children. Early childhood education, as in many other areas, demand a constant attention and dedication in order to maintain or increase the process quality. Several decades of research demonstrate that adequate early childhood education is fundamental to ensure a sound basis for later life. There are several indicators that can be used to assess children involvement, such as Head Posture, Gaze direction. In this work we focus to analyze children head posture and gaze direction to measure their concentration on an involvement.

Human attention needs to be evaluated for different reasons. In this project we'll try to build a system that takes the first step of measuring human concentration by classifying eye blink and eye gaze direction.

1.5 Problem Definition

A general statement of object detection can be described as, "Object detection is the process of finding instances of real-world objects such as faces, bicycles, and buildings in images or videos." Object detection algorithms typically use extracted features and learning algorithms to recognize instances of an object category.

And the classification technique can be stated as, "Classification is a technique where we categorize data into a given number of classes". The main goal of a classification problem is to identify the category/class to which a new data will fall under. Classifier is an algorithm that maps the input data to a specific category.

Eye blinking is partly subconscious fast closing and reopening of the eyelid. In this project, we used classification technique to classify eye blink and also used object detection algorithm to detect pupil.

Now a days, there are several object detection and classification algorithms have been using in every aspect of our lives. Our aim is to integrate and taking both object detection and classification technique under one roof. Furthermore, we want to detect and classify in real-time from a video sequence. To extracting feature, we used Histogram of Oriented Gradients(HOG) [2], to classify eye openness we used Support Vector Machine(SVM) [10], to detect eye we used Viola-Jones object detection algorithm [11], to detect edges of pupil we used Hough Transform [12].

1.6 Objective

Our main objective in this project is to fully understand the procedure behind eye blink classification and gaze direction detection and also integrate the classification and object detection algorithms. Eye blink and gaze direction is hard to measure in real-time. A video stream has lots of image frames, from which feature would be extracted to detect and classify. So, we have to implement in a system that can calculate with better time complexity. Keeping this in mind, we are approaching towards integrating classification technique and object detection.

1.7 Outline

The outline of this project report consists of 6(six) different chapters in total. Starting with chapter 1, where the problem domain and necessary background information is described. Continue with chapter 2 gives an description on the related works and research domains that has already been done in this field. Moving on to chapter 3, our proposal is discussed at length with theoretical and empirical overview along with necessary flow diagram. Then, chapter 4 describes the implementation in details with necessary equations and algorithms. At last, chapter 5 and 6 includes the results of the project and a brief conclusion respectively.

CHAPTER 2

LITERATURE REVIEW

2.1 Research Domain

Scientists and researchers around the globe have be working with the technologies of digital image processing, computer vision, machine learning, data mining. In the field of automation system, there have been a lot of works done by the researchers. But smart solution can't be smart enough if it doesn't come with higher accuracy rate and doesn't work in real-time. For this reason, real-time object detection, feature tracking, classification has been a big challenge. There's a lot of algorithms which works real-time, but there's no limit to increase performance and specifications.

There has been many inter disciplinary research work as well. Which led to many hybrid innovations. Real-time eye blink and gaze detection can provide some extra-ordinary facilities in our day to day life such as automation system, children attention evaluation, smart breaking system in vehicle, smart auto-pilot system which will contribute to decrease accident rates. Some of the research domain are described below:

Fields	**Scenarios of applications (Examples)**
Smart break	Drivers drowsiness occurs accident, smart break can prevent accidents.
Automated vehicle system	Sometimes drivers can be unable to drive for unavoidable reasons, this system can save time and lives.
Preschool children attention	Preschool children needs extra and uninterrupted care, this system can evaluate children's attention.
Smart Auto-pilot in vehicles	Due to tiredness drivers needs rest for a short period of time while driving, smart auto-pilot can greatly help them.
Human attention and expression evaluation.	Evaluating human attention, expression, concentration is a very difficult job since it doesn't have specific patterns.

2.2 Related Works

Among tons of research works, these are closely relatable to our project:

Real-Time Eye Blink Detection using Facial Landmarks [13]

In this research they proposed a real-time algorithm to detect eye blinks in a video sequence from a standard camera is proposed in this paper. Recent landmark detectors, trained on in-the-wild datasets exhibit excellent robustness against a head orientation with respect to a camera, varying illumination and facial expressions. Estimate the landmark positions, extracts a single scalar quantity – eye aspect ratio (EAR) -characterizing the eye opening in each frame.

Finally, an SVM classifier detects eye blinks as a pattern of EAR values in a short temporal window. The simple algorithm outperforms the state-of-the-art results on two standard datasets. Ability of two state-of-the-art landmark detectors to reliably distinguish between the open and closed eye states is quantitatively demonstrated on a challenging in-the-wild dataset and for various face image resolutions. A novel real-time eye blink detection algorithm which integrates a landmark detector and a classifier is proposed. The evaluation is done on two standard datasets achieving state-of-the art results.

Driver Gaze Region Estimation Without Using Eye Movement [3]

In this article research has been done in automated estimation of a driver's visual attention. Vision-based tracking of the eye can provide a good estimate of gaze location. They developed system that extracts facial features and classifies their spatial configuration into six regions in real-time.

This paper shows that spatial configuration of facial landmarks provides sufficient discriminating information to accurately classify driver gaze into six gaze regions. The proposed system achieves an average accuracy of 91.4% at an average decision rate of 11 Hz for an on-road dataset of 50 subjects. Future work of this research is exploring and exploiting inter-person variation with the relationship between eye and head movement. Intra-person variation with the relationship between eye and head movement.

A Framework for Robust Driver Gaze Classification [14]

The key components of this research framework are: Semi-automated annotation of large on-road datasets, offline training of classifiers based on the labeled data and real-time gaze classification that is robust to variable lighting conditions and occlusion that prevent successful detection of the pupil.

In this article, correlation between head and eye movement have shown inter-person variation in the degree to which the head serves as a proxy for gaze. A previous work tested drivers' head movements while looking at the 'road' and the 'center stack' and found that: Drivers' horizontal range of head movements varied (from 5 to 20 degrees) across individuals along with their mean differences of horizontal head angles while looking at the two objects (from 0 to 10 degrees)

The two key aspects of semi-automated annotation approach are: The problem of video annotation into the problem of change detection and a Hidden Markov Model (HMM) is used to model the changes between states in order to predict when a sequence of classifier decisions correspond to a sequence of stable state self-transitions.

Owl and Lizard: Patterns of Head Pose and Eye Pose in Driver Gaze Classification [15]

The main insight of the paper is conveyed through the analogy of an "owl" and "lizard" which describes the degree to which the eyes and the head move when shifting gaze. When the head moves a lot ("owl"), not much classification improvement is attained by estimating eye pose on top of head pose. On the other hand, when the head stays still and only the eyes move ("lizard"), classification accuracy increases significantly from adding in eye pose.

They characterized how the accuracy varies between people, gaze strategies, and gaze regions. This paper investigates the contribution of head pose and eye pose to gaze classification accuracy for different gaze strategies. This paper answer two questions: (1) how much does eye pose contribute and (2) how can the inter-user accuracy variation be explained?

For the former, we show that eye pose adds a 5.4% increase in average accuracy (from 89.2% to 94.6%) with an effective average rate of 1.3 decisions per second. For the latter, we propose an "owlness" metric that decomposes gaze into head movement and eye movement and computes the relative magnitude of each.

Head Posture Detection for Measuring Preschoolers Concentration [16]

The work described in this paper focus on how modern image processing technology can provide a valuable aid to kindergarten teachers, helping them in the task of registering observations.

In this context, head posture is automatically detected and measured, and time is recorded. Although easy for a human to interpret the orientation and movement of the human head, it is a challenge to computers. Of course there is always subjectivity in this kind of observation, but we hope that this tool can contribute to help teachers to make informed and critical judgments about the quality of teaching and learning they offer.

The use of observation instruments that stimulate reflection on the educational activity are an important support the kindergarten teacher's decisions. The child's involvement scale allows them to explore the issues related to the quality of learning experiences the possibilities provided by the multi-functionality of space and daily routine as well as the type of interactions established.

Enabling environment created by the intentional act of kindergarten teachers constitute itself as a key element in the involvement of children. If the observation data indicate that children are not involved, adults should reflect critically about the aspects to improve.

2.3 Research Gain

We have most definitely earned insight through analyzing these research papers and many more that has been excluded from this report. By learning through this, one thing made certain sense that there needs to be an evaluation among all the classification methods. There has been research regarding evaluation of these classification techniques. But many of these papers does not include the algorithms that we are to use. Which makes our project contribute to the broader efficiency.

Without proper research, our project would have been incomplete. It's these researches that helped us pave the way towards our goal. As our project is based on integrating existent system and techniques, the data and information collected through research papers are considered the main contributor of our project.

CHAPTER 3

DESCRIPTION OF THE PROPOSAL

3.1 Comprehensive Description

Comprehensive description of this report includes the actual ideology behind the project in detail. In classification and object detection, many review papers have been published by doing comparisons between quite a few feature extractor techniques.

Our research focuses on integrating SVM classifier, Viola-Jones algorithm and Hough Transform. For this research, we have set the constraints as to be using HOG for feature extraction and real-time training for classification.

3.2 Flow Diagram

Flow diagram of our project describes the procedure and its sequence. The process follows these steps:

1. Start with a video stream and takes image frames at real-time.
2. Select the Region of Interest(ROI).
3. Detect face in the region of interest.
4. Detect the eye pair candidate.
5. Verify the eye pair candidate.
6. If successful, goes to HOG feature extraction and Viola-Jones object detection.
7. Extracted HOG features goes to SVM classifier model.
8. SVM classifies eye blink.
9. Viola-jones detects one eye from the eye pair candidate.
10. Hough transform creates a circle around the edges of the pupil.
11. Metric distance is measured to evaluate gaze direction.
12. If not successful at step 5 system goes to step 1.
13. Otherwise system ends.

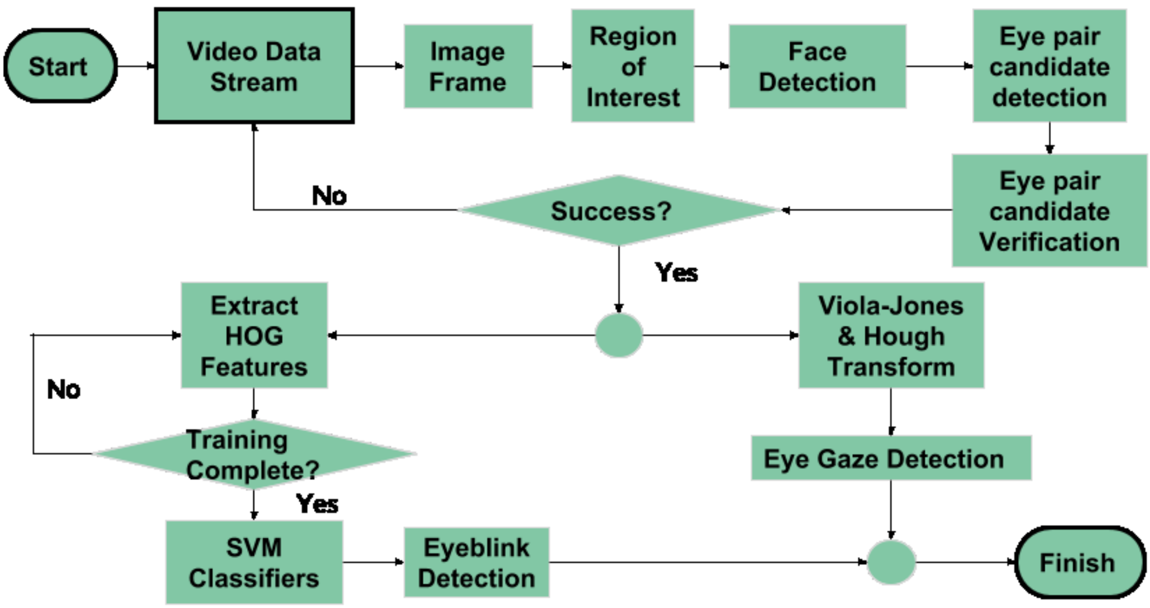

Figure 1 : Flow diagram of the proposed system

3.3 PROPOSAL OVERVIEW

3.3.1 Theoretical Overview

From theoretical aspect of our project, the proposal includes classifiaction techniques and object detection and its processes. It also includes HOG as a feature extractor and Hough Transform to detect the edges of pupil. We used these algorithms and techniques:

1. Histogram of Oriented Gradients(HOG). [2]
2. Support Vector Machine (SVM). [10]
3. Viola-Jones Algorithm. [11]
4. Hough Transform. [12]

3.3.2 Empirical Overview

Along with theories behind our project, the empirical view is an important factor for evaluation of the classification and object detection methods.

In short, the proposal overview of our project depicts on the fundamental working needed in order to proceed with the work. The project only succeeds by doing both theoretical and practical evaluation of performance measure. The verification of our project depends on the theoretical aspect and validation depends on the empirical aspect of it.

CHAPTER 4

IMPLEMENTATION

4.1 Image Acquisition

Image acquisition is one of the earliest requirements of any research. To deal with images and before analyzing them the most important thing is to capture the image. This is called as Image Acquisition. Image Acquisition is achieved by suitable camera. For our project, we have depended on the WEBCAM of our PC. From the video stream we'll extract image frames as our input image. Since the system evaluates at real-time, single image can't be used.

4.2 Feature Extraction

Histogram Of Oriented Gradients(HOG) [2]

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

This method requires filtering the color or intensity data of the image with the following filter kernels: [-1 0 1] and $[-1\ 0\ 1]^T$.

Steps in HOG:

1. Gradient computation.
2. Orientation binning.
3. Descriptor blocks.
4. Block normalization.
5. Object recognition.

4.3 Classification Technique

In our project we used Support Vector Machine Classifier as our classification technique. This step permits the classification of the feature vector of the person to recognize. This treatment requires the introduction of a comparison algorithm or classification which provides at its output a score of similarity or distance between this characteristic vector and the reference feature vectors of the image frame. This score is compared subsequently to a decision threshold fixed in advance for provide a final decision on identity.

4.3.1 Support Vector Machine(SVM) Classifier:

Support Vector Machines (SVMs) are research topic in machine learning community, creating a similar enthusiasm as in Artificial Neural Network. Library for Support Vector Machines (LIBSVM) has been actively used to develop this package since 2000. The aim is to help users easily apply the SVM in their applications. The SVM tries to find the optimal separating hyper plane that maximizes the margin of separation in order to minimize the risk of misclassification in both training samples and the unseen data in the test set. The algorithm is defined by a weighted combination of small subset of the training vectors, called support vectors. The class separating hyper plane is chosen to minimize the expected classification. Estimating the optimal hyper plane is equivalent to solving a linearly constrained quadratic programming problem. The SVM algorithm is applied for classification of facial expression characteristics. It can be considered a new paradigm to train polynomial functions, radial basis function classifiers, neural networks, and operate on another induction principle called structural risk minimization, which aims at minimizing an upper bound on the expected generalization error. Most methods for training a classifier are based on minimizing the training error (i.e. empirical risk).

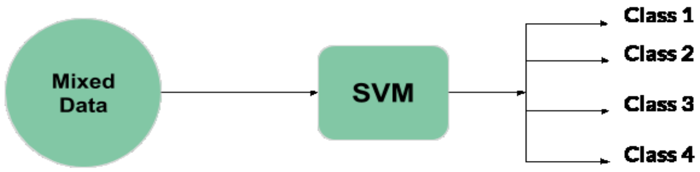

Figure 2 SVM with mixed data

SVM is currently attracting much attention in the community of machine learning, which proves their gain in popularity and use machine learning in many applications such as pattern recognition (handwritten scriptures, faces), text categorization (classification of emails, web pages classification), medical diagnosis (risk assessment of cancer, cardiac arrhythmia detection). [17]

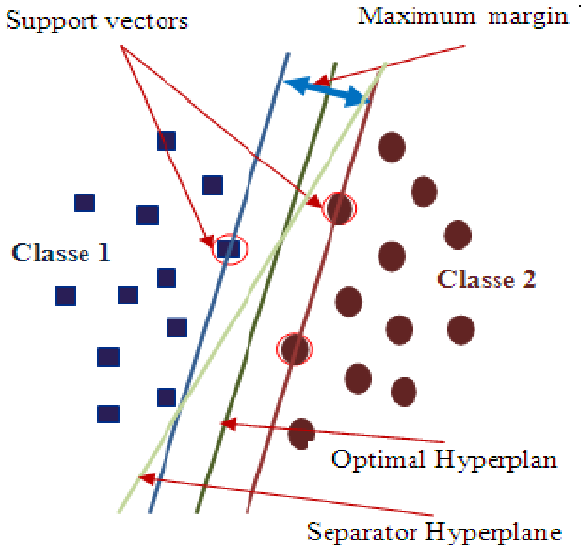

Figure 3 Basic Illustration of SVM

The main objective of SVM is to find a decision boundary that separates the data points of two different classes. This boundary is called a separator, must be a hyperplane. In general, there may be multiple hyperplanes possible separators between the two classes. However, it made a particular choice among all possible separators, it seeks Optimal Separating hyperplane. To determine the latter, closest OSH uses only data points (the points of the boundary between the two classes of data) from the total set of learning, these points are called support vectors (support vectors). The distance between these points is called margin. It is this distance that we must maximize (maximum margin). For details on obtaining certain mathematical formulas necessary for the implementation of the SVM or an understanding of SVM, the reader may refer to the memory of the thesis. [18]

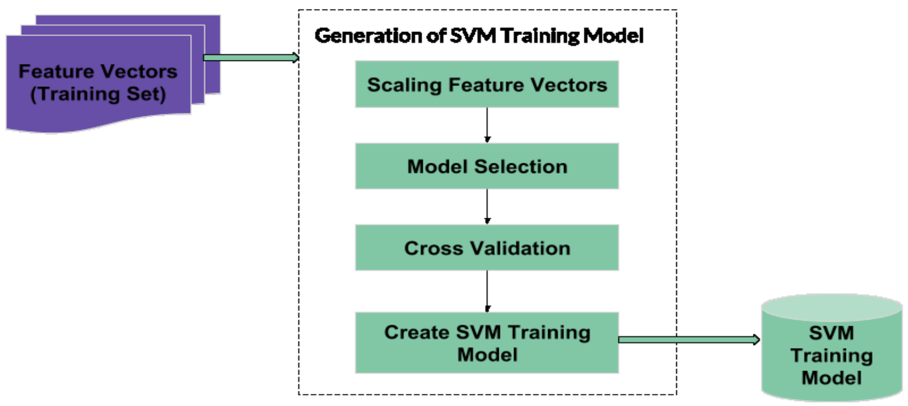

Figure 4 Generation of SVM Training Model

4.3.2 Classification for Eye Blink Detection

Steps in classification for eye blink detection:

1. Image acquisition.
2. Training with positive class (closed eyes) and negative class(opened eyes).
3. HOG feature extraction.
4. Send to SVM classifier model.
5. Classification: Positive class or Negative class.
6. Go to image acquisition.

Diagram:

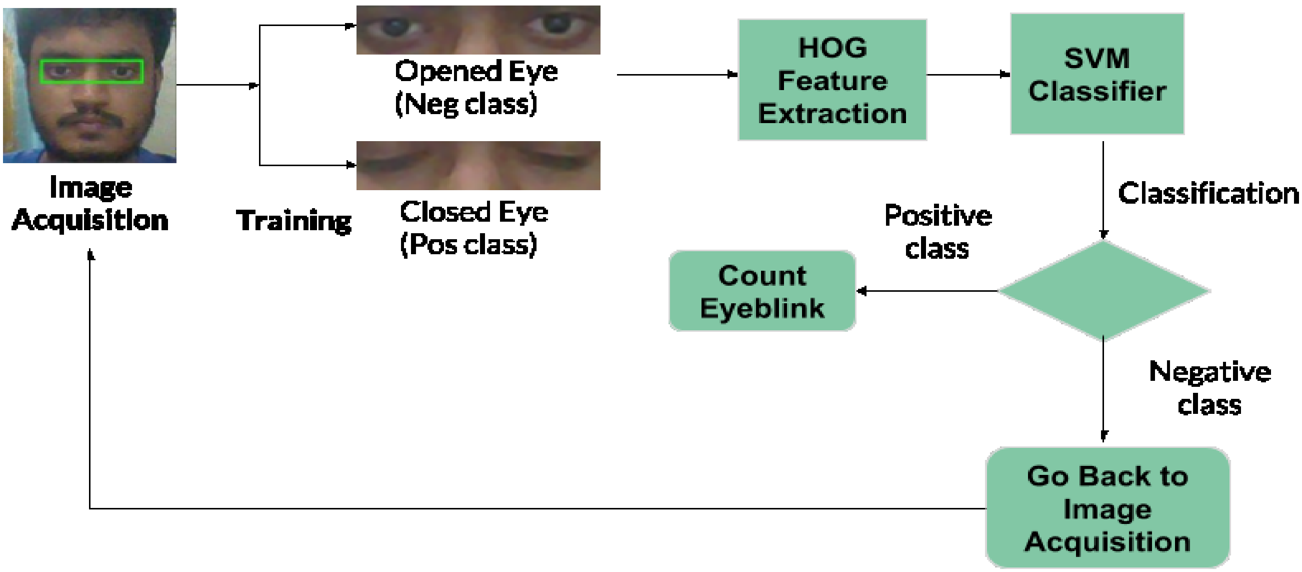

Figure 5 Classification of eye blink detection

4.4 Viola-Jones Object Detection Algorithm

The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. [11]

The algorithm has four stages:

1. Haar Feature Selection
2. Creating an Integral Image
3. Adaboost Training
4. Cascading Classifiers

A simple framework for cascade training is given below:

- f = the maximum acceptable false positive rate per layer.
- d = the minimum acceptable detection rate per layer.
- Ftarget = target overall false positive rate.
- P = set of positive examples.
- N = set of negative examples.

```
F(0) = 1.0; D(0) = 1.0; i = 0

while F(i) > Ftarget
   increase i
   n(i) = 0; F(i)= F(i-1)

   while F(i) > f × F(i-1)
     increase n(i)
     use P and N to train a classifier with n(I) features using AdaBoost
     Evaluate current cascaded classifier on validation set to determine F(i) and D(i)
     decrease threshold for the ith classifier
       until the current cascaded classifier has a detection rate of at least d × D(i-1) (this also affects F(i))
     N = ∅
     if F(i) > Ftarget then
       evaluate the current cascaded detector on the set of non-face images
       and put any false detections into the set N.
```

Diagram:

Flow Diagram:

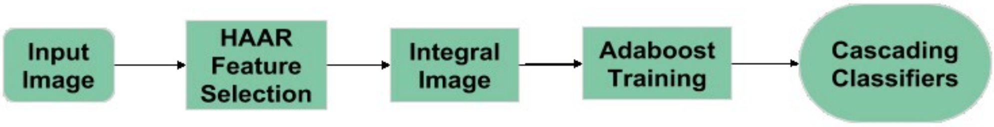

Figure 6 Viola-jones algorithm for one eye detection

4.5 Hough Transform Feature Extraction

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. [12]

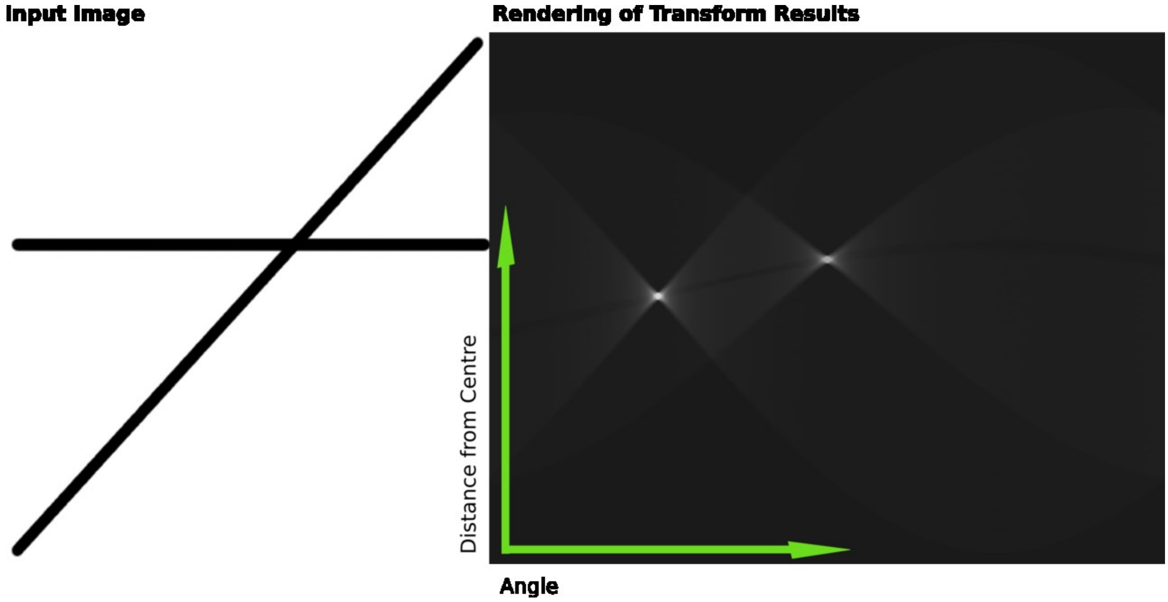

Figure 7 Hough Transform Edge Detection

<u>HOUGH TRANSFORM PUPIL SHAPE DETECTION:</u>

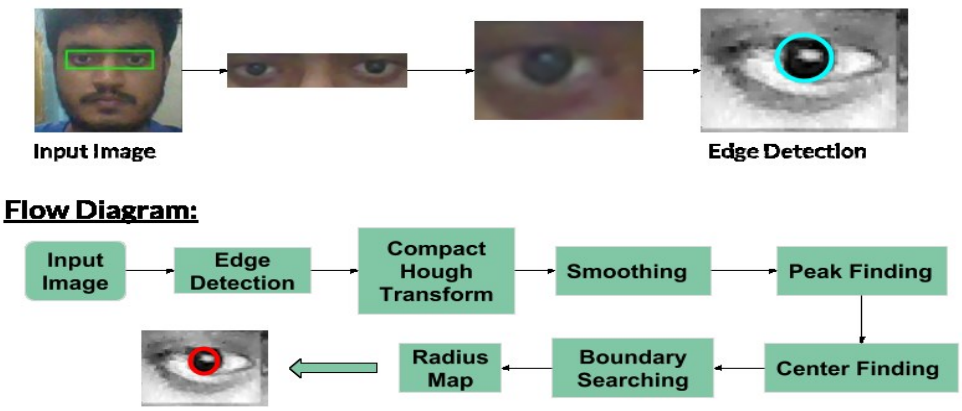

Figure 8 Pupil edge detection with hough transform

4.6 Software Specification

4.6.1 MATLAB

MATLAB (matrix laboratory) is a multi-paradigm mathematical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allowing matrix handlings, plotting of utilities and using of existing of library, working UI, and it supports programming languages such as: C, C++, C#, Java, Fortran and Python. [19]

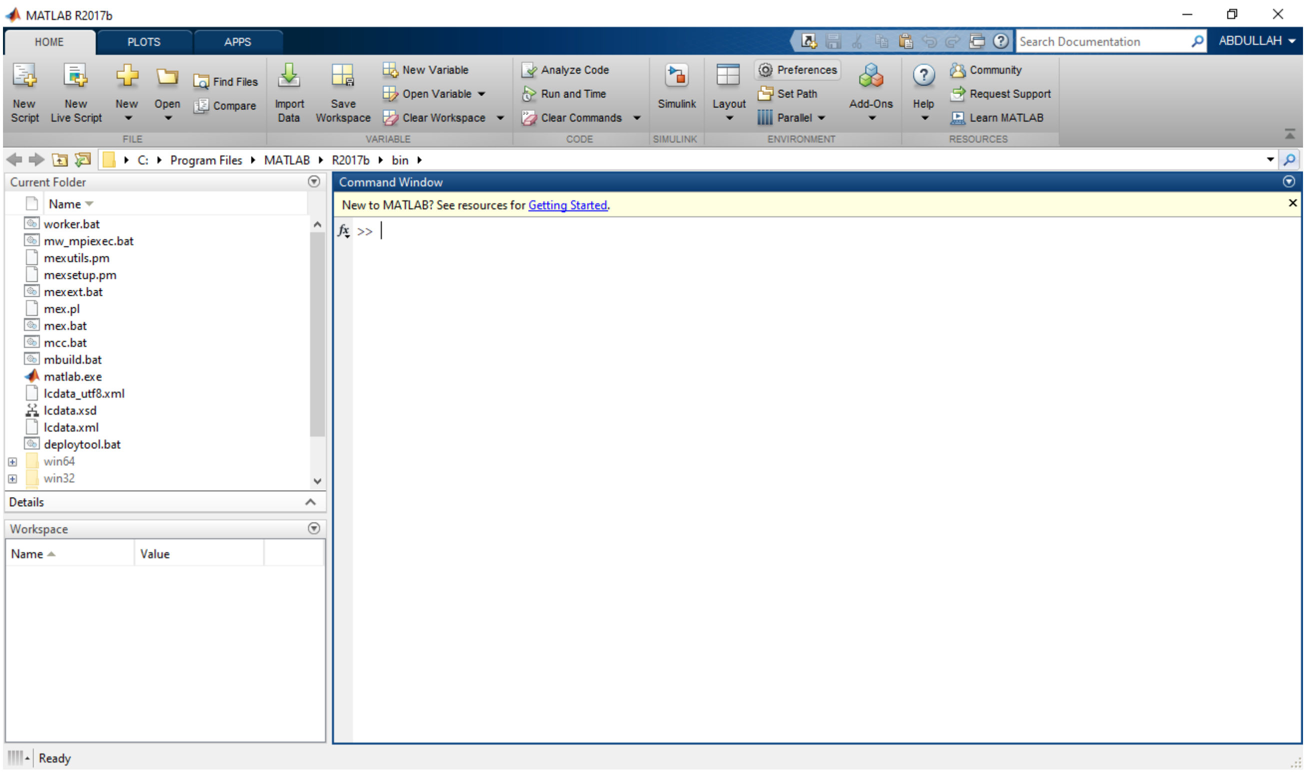

Figure 9 MATLAB Interface

CHAPTER 5

RESULTS AND DISCUSSION

5.1 System Demo

When we run the system it opens several windows. The window list is given below:

1. Capturing and training tool.
2. Deployable video player.
3. Zoomed deployable video player.
4. Capturing training data.
5. Gaze direction display.

These figures are as follows:

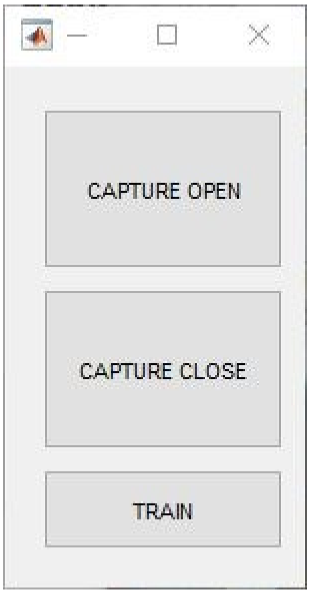

Figure 10 Capturing and training data

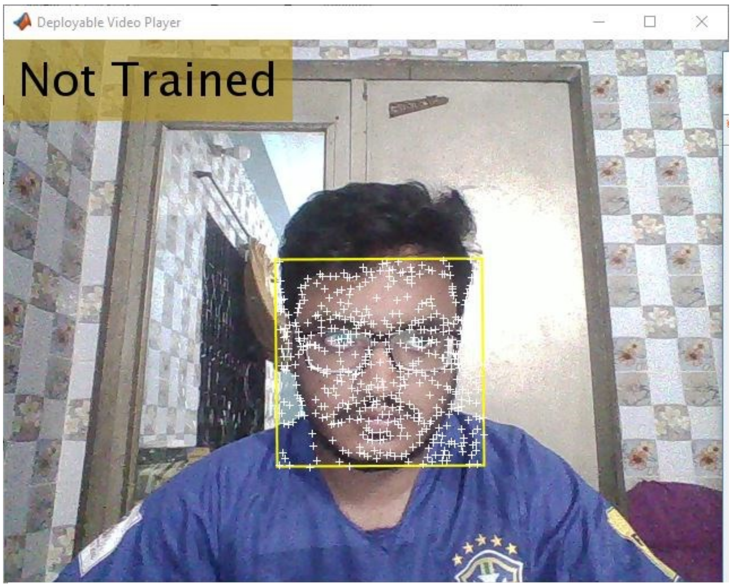

Figure 11 Deployable video player

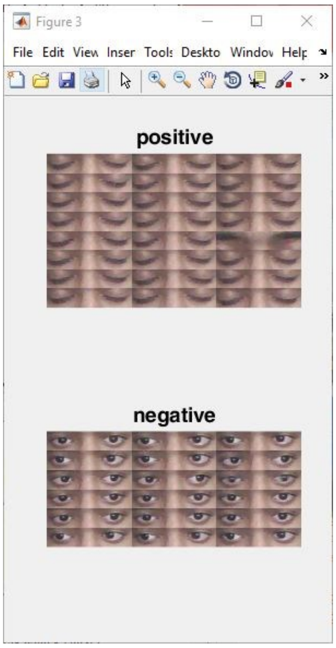

Figure 12 Capturing Training Data

Figure 13 Counting eye blink detection

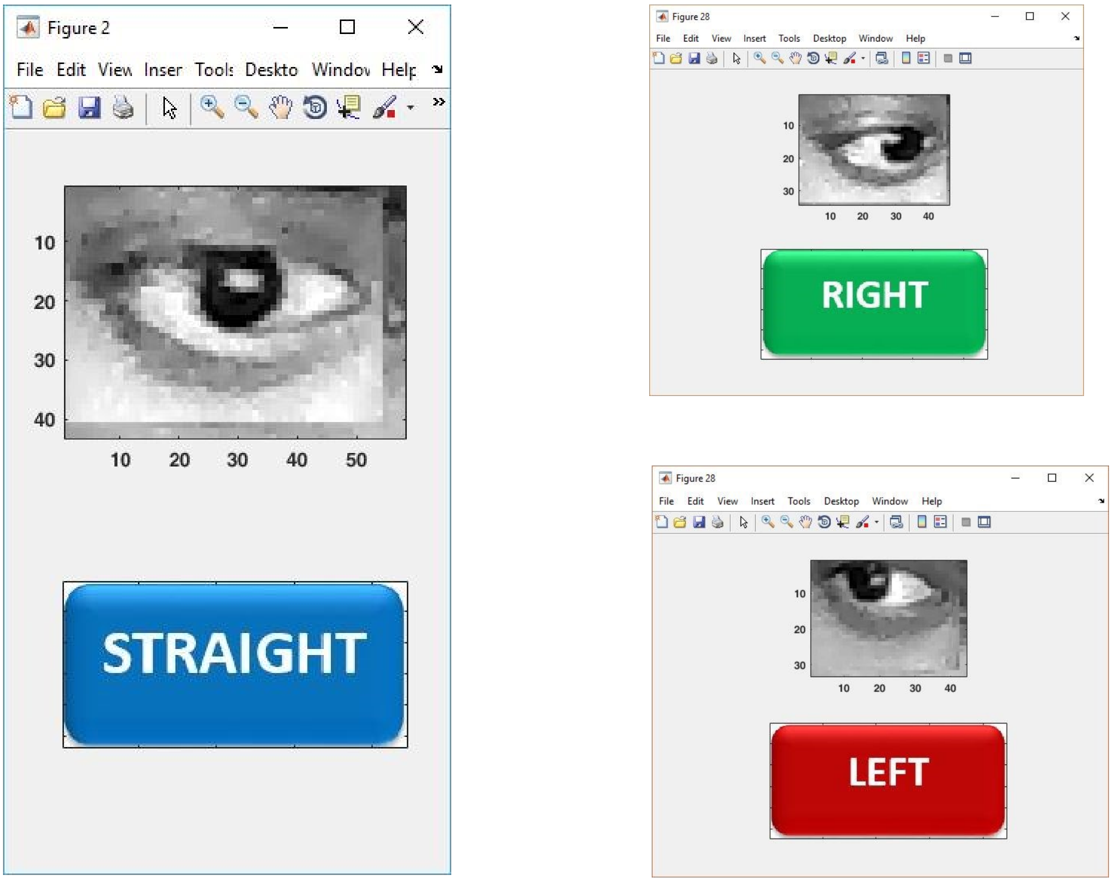

Figure 14 Eye gaze detection

CHAPTER 6

CONCLUSION

6.1 Summery

In this project we wanted to build a real-time eye blink and eye gaze direction detection system from a video sequence. Classification based eye blink detection has been done in this project.

Histogram of Oriented Gradient (HOG) is used as a feature descriptor. Eigen face features are detected in the Region of Interest (ROI). From rectified face eye pair are detected. Support vector machine (SVM) is used as classifier to classify closed eyes as positive class and opened eyes as negative class as they are binary classes.

With the help of the HAAR feature selection method in Viola-Jones object detection algorithm, Adaboost training has been implemented to integrate images to detect objects such as one single eye. Hough transform edge detection has been implemented to detect the edges of the pupil in a single eye and draw a circle around the pupil. As detecting the pupil, metric distances from the eye-edges were measured to evaluate gaze direction.

Our main objective in this project was to fully understand the procedure behind eye blink classification and gaze direction detection and also integrate the classification and object detection algorithms.

6.2 Future Works

Gaze region estimation along with head posture will be more precise to estimate human behavior. Along with facial landmarks eye blink and eye gaze can classify human expressions. Semi-automated vehicle intelligent system can be generated. Smart braking system along with smart auto-pilot system can be introduced. Head posture detection can be implemented to classify human concentration.

References

[1] T. Danisman, I. Bilasco, C. Djeraba, and N. Ihad-dadene., "Drowsy driver detection system using eye blink patterns.," *Machine and Web Intelligence.,* October, 2010.

[2] "Histogram of oriented gradients," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.

[3] Lex Fridman, Philipp Langhans, Joonbum Lee and Bryan Reimer, "Driver Gaze Region Estimation Without Using Eye Movement," *IEEE,* March, 2016.

[4] T. Soukupov´a, "Eye Blink Detection Using Facial Landmarks," *Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University.,* 2016.

[5] P. Chakravorty, "What Is a Signal? [Lecture Notes]," *IEEE Signal Processing Magazine,* Vols. 35, no. 5, pp. 175-177, September, 2018.

[6] Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Digital_image_processing.

[7] S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, and D. J. Ramsey, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data," Tech. Report, 2016.

[8] Lex Fridman, Philipp Langhans, Joonbum Lee, and Bryan Reimer, "Driver Gaze Region Estimation Without Using Eye Movement," *IEEE,* March, 2016.

[9] Coughlin Joseph F, Reimer Bryan, and Mehler Bruce., "Monitoring, managing, and motivating driver safety and well-being," *IEEE Pervasive Computing,* vol. 10(3), 2011.

[10] "Support-vector machine," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Support-vector_machine.

[11] "Viola–Jones object detection framework," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework.

[12] "Hough transform," Wikipedia, the free encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Hough_transform.

[13] Tereza Soukupova and Jan Cech, "Real-Time Eye Blink Detection using Facial Landmarks," *Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague.,* February, 2016.

[14] Fridman, L., Lee, J., Reimer, B., and Mehler, B., "A Framework for Robust Driver Gaze Classification," *SAE Technical,* 2016.

[15] L. Fridman, Joonbum Lee, Bryan Reimer, and Trent Victor, "Owl and Lizard: Patterns of Head Pose and Eye Pose in Driver Gaze Classification," *IEEE,* 2016.

[16] CRISTINA MESQUITA-PIRES, RUI PEDRO LOPES, "Head Posture Detection for Measuring Preschoolers Concentration," *Polytechnic Institute of Bragança (Portugal).*

[17] [Online]. Available: https://www.researchgate.net/publication/295859899.

[18] El-Dahshan, E. S. A., Hosny, T., & Salem, A. B. M. (, "Hybrid Intelligent Techniques for MRI Brain Images Classification. Digital Signal Processing,," 2010.

[19] [Online]. Available: https://www.mathworks.com/products/matlab.html.

APPENDIX

Code for eye blink and gaze detection:

```
% Tracking by recognizing faces (using Kanade-Lucas-Tomasi (KLT))
% Detect closed eyes from position of face
% We use mechanics SVM (Support Vector Machine) for inspection
% Statistics Toolbox is required

%% initialize
clear all; close all; clc; imaqreset;

%% Image preservation folder
imdir = 'dbim';
[~,~,~] = rmdir(imdir,'s');
imdirClosed = [imdir filesep 'closed'];
imdirOpened = [imdir filesep 'opened'];
[~,~,~] = mkdir(imdir);
[~,~,~] = mkdir(imdirClosed);
[~,~,~] = mkdir(imdirOpened);

%% Video aquisition (Object Definition)
cam = webcam();


right=imread('RIGHT.jpg');
left=imread('LEFT.jpg');
straight=imread('STRAIGHT.jpg');

% Display object
videoPlayer = vision.DeployableVideoPlayer();
videoPlayerZoom = vision.DeployableVideoPlayer();

%face and eye detector
faceDetector = vision.CascadeObjectDetector();
eyesDetector = vision.CascadeObjectDetector('EyePairBig');

% Point Tracker Object (Tracking Face)
pointTracker = vision.PointTracker('MaxBidirectionalError', 2);


runLoop = true;
numPts = 0;
frameCount = 0;
eyeFlag = false;
eyeCount =  int32(0);
time = 0;
numClosed = 0;
numOpened = 0;
isCaptureClosed=false;
isCaptureOpened=false;
isTrain=false;

%% Capture/Train
sz = get(0,'ScreenSize');
figure('MenuBar','none','Toolbar','none','Position',[60 sz(4)-500 150 260])
uicontrol('Style', 'pushbutton', 'String', 'CAPTURE OPEN',...
    'Position', [20 160 120 80],...
```

```matlab
    'Callback', 'isCaptureOpened=true;');
uicontrol('Style', 'pushbutton', 'String', 'CAPTURE CLOSE',...
    'Position', [20 70 120 80],...
    'Callback', 'isCaptureClosed=true;');
uicontrol('Style', 'pushbutton', 'String', 'TRAIN',...
    'Position', [20 20 120 40],...
    'Callback', 'isTrain=true;');
k = figure('Position', [1000 30 300 500]);
%training image display
h = figure('Position', [200 30 300 500]);
subplot(2,1,1);
set(gca,'xtick',[],'ytick',[],'Xcolor','w','Ycolor','w')
title('Positive Class(Eyes Closed)','FontSize',14);
subplot(2,1,2);
set(gca,'xtick',[],'ytick',[],'Xcolor','w','Ycolor','w')
title('Negative Class(Eyes Opened)','FontSize',14);


while (runLoop)
    %capture image data
    videoFrameOrg = snapshot(cam);

    videoFrame = videoFrameOrg;
    videoFrameGray = rgb2gray(videoFrameOrg);

    % Tracking points
    if numPts < 10
        % Face Detection
        bbox = faceDetector.step(videoFrameGray);

        if ~isempty(bbox)
            % Inspect the tracker point from the detected area
            points = detectMinEigenFeatures(videoFrameGray, 'ROI', bbox(1, :));

            % Point Tracker
            xyPoints = points.Location;
            numPts = size(xyPoints,1);
            release(pointTracker);
            initialize(pointTracker, xyPoints, videoFrameGray);

            % save points
            oldPoints = xyPoints;

            % Convert to co-ordinator
            bboxPoints = bbox2points(bbox(1, :));

                    % Convert to a vector of the form shown on the right [x1 y1 x2 y2 x3 y3 x4 y4]
                    bboxPolygon = reshape(bboxPoints', 1, []);

                    % Display boundary area
                    videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

                    % Display detected corners
                    videoFrame = insertMarker(videoFrame, xyPoints, '+', 'Color', 'white');
                end

```

```
        else
            %Tracking Mode
            [xyPoints, isFound] = step(pointTracker, videoFrameGray);
            visiblePoints = xyPoints(isFound, :);
            oldInliers = oldPoints(isFound, :);

            numPts = size(visiblePoints, 1);

            if numPts >= 10
                % Geometric transformation of old points
                [xform, oldInliers, visiblePoints] = estimateGeometricTransform(...
                    oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);

                % Convert to a ticket using a row
                bboxPoints = transformPointsForward(xform, bboxPoints);

                % Convert to a vector of the form shown on the right [x1 y1 x2 y2 x3 y3 x4 y4]
                bboxPolygon = reshape(bboxPoints', 1, []);

                %  Display boundary area
                videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

                % Display detected corners
                videoFrame = insertMarker(videoFrame, visiblePoints, '+', 'Color', 'white');

                % reset points
                oldPoints = visiblePoints;
                setPoints(pointTracker, oldPoints);

            end

            % Fitting the face part into a square (for both eye examination)
            zoomPoints = [1 1; 200 1; 200 200; 1 200];
            T = fitgeotrans(bboxPoints, zoomPoints,'projective');
            rectifiedFace = imwarp(videoFrameOrg, T, 'OutputView', imref2d([200 200]));

            % Detect both eyes from the image of the face
            bboxEyes = step(eyesDetector, rectifiedFace);

            if ~isempty(bboxEyes) % Was I able to find both eyes?
                % Fit the region of both eyes detected into a rectangle
                bboxEyesPoints = bbox2points(bboxEyes(1,:));
                zoomPoints = [1 1; 440 1; 440 100; 1 100];
                T = fitgeotrans(bboxEyesPoints, zoomPoints,'projective');
                videoFrameEyes = imwarp(rectifiedFace, T, 'OutputView', imref2d([100 440]));

                if (isCaptureClosed || isCaptureOpened)
                    % Save closed eyes to file
                    if (isCaptureClosed)
                        numClosed = numClosed+1;
                        imwrite(videoFrameEyes, [imdirClosed filesep 'frame' num2str(numClosed) '.png']);
                    end

```

```
        % Save open eyes in a file
        if (isCaptureOpened)
            numOpened = numOpened+1;
            imwrite(videoFrameEyes, [imdirOpened filesep 'frame' num2str(numOpened) '.png']);
        end

        % Specify a set of training images
        posSets = imageSet(fullfile(imdir,'closed'));
        negSets = imageSet(fullfile(imdir,'opened'));
        figure(h);
        subplot(2,1,1);
        if ~isempty(posSets.ImageLocation)
            montage(posSets.ImageLocation);% pos image
        end
        title('Positive Class(Eyes Closed)','FontSize',14);
        subplot(2,1,2);
        if ~isempty(negSets.ImageLocation)
            montage(negSets.ImageLocation);% neg image
        end
        title('Negative Class(Eyes Opened)','FontSize',14);
        isCaptureClosed = false;
        isCaptureOpened = false;
    end


    if exist('svmModel','var')
        img = imresize(videoFrameEyes, [20 88]);       % Changing image size
        % Extract HOG features
        testFeatures = extractHOGFeatures(img,'CellSize',cellSize);
        % Predictions by passing features to classifiers
        preEyeFlag = eyeFlag;
        eyeFlag = predict(svmModel, testFeatures);
        if ~preEyeFlag && eyeFlag
            eyeCount = eyeCount + 1;
        end
    else
        preEyeFlag = false;
    end

    % Display boundary showing detected eye pupil
    rectifiedFace = insertShape(rectifiedFace, 'Rectangle', bboxEyes, 'LineWidth', 3, 'Color', 255*[eyeFlag ~eyeFlag 0]);
end


if isTrain
    isTrain = false;

    % Specify a set of training images
    trainingSets = imageSet(imdir, 'recursive');

    % Show training image set
    figure(h);
    subplot(2,1,1);montage(trainingSets(1).ImageLocation);title('positive','FontSize',14);   % pos image
    subplot(2,1,2);montage([trainingSets(2:end).ImageLocation]);title('negative','FontSize',14); % neg image
```

```
                    % labeling pos image as true, neg image as false
                    trainingLabels    = false(sum([trainingSets.Count]),1);
                    trainingLabels(1:trainingSets(1).Count) = true; %Specify (closed folder) as pos
                    trainingLabels(trainingSets(1).Count+1:sum([trainingSets(2:end).Count])) = false; %Specify (open folder) as neg

                    % Use cell size of 4x4
                    cellSize = [4 4];

                    % In order to calculate hogFeatureSize in advance, only one HOG extraction
                    img = read(trainingSets(1), 1);
                    img = imresize(img, [20 88]);
                    [hog_4x4, vis4x4] = extractHOGFeatures(img,'CellSize',cellSize);
                    hogFeatureSize = length(hog_4x4);

                    % trainingFeatures Prearrange the array to store
                    trainingFeatures  = zeros(sum([trainingSets.Count]),hogFeatureSize,'single');

                    % Extract HOG feature value from all tracing image
                    k = 1;
                    for index = 1:numel(trainingSets)   % 1=>pos, 2=>neg
                        for i = 1:trainingSets(index).Count
                            img = read(trainingSets(index), i);  % Reading of training images
                            img = imresize(img, [20 88]);
                            trainingFeatures(k,:) = extractHOGFeatures(img,'CellSize',cellSize); % Feature amount extraction
                            k = k+1;
                        end
                    end

                    % Support for classifier of support vector machine (SVM) (for fitcsvm() function)
                    svmModel = fitcsvm(trainingFeatures, trainingLabels);
                end

                % Display the screen
                step(videoPlayerZoom, rectifiedFace);
            end

            % Model display
            if exist('svmModel', 'var')
                % Displaying blink detection mode
                videoFrame = insertText(videoFrame, [0 0], 'Running eyeblink detection', 'FontSize', 40, 'BoxColor', [30 180 160]);
                % Eye close count
                videoFrame = insertText(videoFrame, [0 50],  ['Eye Closed ' num2str(eyeCount,'%d') ' times'], 'FontSize', 20, 'BoxColor', [140 30180]);
            else
                % Displaying not trained
                videoFrame = insertText(videoFrame, [0 0], 'Not Trained', 'FontSize', 40, 'BoxColor', [180 150 30]);
            end


```

```
            step(videoPlayer, videoFrame);

            % Set runLoop to false when the video player is closed
            runLoop = isOpen(videoPlayer);

            drawnow;


            img = flip(videoFrameGray, 2); % Flips the image horizontally

            bbox = step(faceDetector, img); % Creating bounding box using faceDetector

            if ~ isempty(bbox)  %if face exists
                biggest_box=1;
                for i=1:rank(bbox) %find the biggest face
                    if bbox(i,3)>bbox(biggest_box,3)
                        biggest_box=i;
                    end
                end
                faceImage = imcrop(img,bbox(biggest_box,:)); % extract the face from the image
                bboxeyes = step(eyesDetector, faceImage); % locations of the eyepair using faceDetector

                if ~ isempty(bboxeyes)  %check it eyepair is available

                    biggest_box_eyes=1;
                    for i=1:rank(bboxeyes) %find the biggest eyepair
                        if bboxeyes(i,3)>bboxeyes(biggest_box_eyes,3)
                            biggest_box_eyes=i;
                        end
                    end

                    bboxeyeshalf=[bboxeyes(biggest_box_eyes,1),bboxeyes(biggest_box_eyes,2),bboxeyes(biggest_box_eyes,3)/3,bboxeyes(biggest_box_eyes,4)];   %resize the eyepair width in half

                    eyesImage = imcrop(faceImage,bboxeyeshalf(1,:));    %extract the half eyepair from the face image
                    eyesImage = imadjust(eyesImage);    %adjust contrast

                    r = bboxeyeshalf(1,4)/4;
                    [centers, radii, metric] = imfindcircles(eyesImage, [floor(r-r/4) floor(r+r/2)], 'ObjectPolarity','dark', 'Sensitivity', 0.93); % Hough Transform
                    [M,I] = sort(radii, 'descend');

                    eyesPositions = centers;
                    figure(k);
                    subplot(2,1,1),subimage(eyesImage); hold on;

                    viscircles(centers, radii,'EdgeColor','b');

                    if ~isempty(centers)
                        pupil_x=centers(1);
                        disL=abs(0-pupil_x);    %distance from left edge to center point
                        disR=abs(bboxeyes(1,3)/3-pupil_x);%distance from right edge to center point
                        subplot(2,1,2);
                        if disL>disR+16
```

```
                        subimage(right);
                    else if disR>disL
                        subimage(left);
                        else
                           subimage(straight);
                        end
                    end

                end
            end
        end
        set(gca,'XtickLabel',[],'YtickLabel',[]);
        hold off;
    end
```

THANK YOU