

12-2015

Designing Eye Tracking Algorithm for Partner-Assisted Eye Scanning Keyboard for Physically Challenged People

Zeenat S. Al-Kassim

Follow this and additional works at: https://scholarworks.uaeu.ac.ae/all_theses

Part of the [Engineering Commons](#)

Recommended Citation

Al-Kassim, Zeenat S., "Designing Eye Tracking Algorithm for Partner-Assisted Eye Scanning Keyboard for Physically Challenged People" (2015). *Theses*. 197.

https://scholarworks.uaeu.ac.ae/all_theses/197

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarworks@UAEU. It has been accepted for inclusion in Theses by an authorized administrator of Scholarworks@UAEU. For more information, please contact fadl.musa@uaeu.ac.ae.

United Arab Emirates University

College of Engineering

Department of Electrical Engineering

DESIGNING EYE TRACKING ALGORITHM FOR PARTNER-
ASSISTED EYE SCANNING KEYBOARD FOR PHYSICALLY
CHALLENGED PEOPLE

Zeenat S. Al-Kassim

This thesis is submitted in partial fulfilment of the requirements for the degree of
Master of Science in Electrical Engineering

Under the Supervision of Dr. Qurban Ali

December 2015

Declaration of Original Work

I, Zeenat S. Al-Kassim, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled “*Designing Eye Tracking Algorithm for Partner-Assisted Eye Scanning Keyboard for Physically Challenged People*”, hereby, solemnly declare that this thesis is an original research work that has been done and prepared by me under the supervision of Dr. Qurban Ali, in the College of Engineering at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student’s Signature _____

Date _____

Copyright © 2015 Zeenat S. Al-Kassim
All Rights Reserved

Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

- 1) Advisor (Committee Chair): Qurban Ali

Title: Associate Professor

Department of Electrical Engineering

College of Engineering

Signature _____ Date _____

- 2) Member: Nabil Bastaki

Title: Assistant Professor

Department of Electrical Engineering

College of Engineering

Signature _____ Date _____

- 3) Member (External Examiner): George Bebis

Title: Professor

Department of Computer Science and Engineering

Institution: University of Nevada, Reno, USA

Signature _____ Date _____

This Master Thesis is accepted by:

Dean of the College of Engineering: Professor Mohsen Sherif

Signature _____ Date _____

Dean of the College of the Graduate Studies: Professor Nagi T. Wakim

Signature _____ Date _____

Copy ____ of ____

Abstract

The proposed research work focuses on building a keyboard through designing an algorithm for eye movement detection using the partner-assisted scanning technique. The study covers all stages of gesture recognition, from data acquisition to eye detection and tracking, and finally classification. With the presence of many techniques to implement the gesture recognition stages, the main objective of this research work is implementing the simple and less expensive technique that produces the best possible results with a high level of accuracy. The results, finally, are compared with similar works done recently to prove the efficiency in implementation of the proposed algorithm. The system starts with the calibration phase, where a face detection algorithm is designed to detect the user's face by a trained support vector machine. Then, features are extracted, after which tracking of the eyes is possible by skin-colour segmentation. A couple of other operations were performed. The overall system is a keyboard that works by eye movement, through the partner-assisted scanning technique. A good level of accuracy was achieved, and a couple of alternative methods were implemented and compared. This keyboard adds to the research field, with a new and novel combination of techniques for eye detection and tracking. Also, the developed keyboard helps bridge the gap between physical paralysis and leading a normal life. This system can be used as comparison with other proposed algorithms for eye detection, and might be used as a proof for the efficiency of combining a number of different techniques into one algorithm. Also, it strongly supports the effectiveness of machine learning and appearance-based algorithms.

Keywords: Eye detection, eye tracking, face detection, skin colour segmentation, support vector machines, artificial neural networks, image processing, partner assisted scanning, RGB colour space

Title and Abstract (in Arabic)

تصميم لوحة المفاتيح بواسطة تحريك العين باستخدام مسح ضوئي بمساعدة شريك للمعاقين جسديا

الملخص

يركز العمل البحثي المقترح على بناء لوحة المفاتيح من خلال تصميم خوارزمية للكشف عن حركة العين باستخدام تقنية المسح الضوئي بمساعدة شريك (Partner-assisted Scanning). وتغطي الدراسة جميع مراحل التعرف على الإيماءات (Gestures)، من الحصول على البيانات لكشف العين والتتبع، وأخيرا تصنيف. مع وجود العديد من التقنيات لتنفيذ مراحل التعرف على الإيماءات (Gesture Recognition) ، فإن الهدف الرئيسي من هذا العمل البحثي هو تنفيذ تقنية بسيطة وأقل تكلفة التي تنتج أفضل النتائج الممكنة مع مستوى عال من الدقة. النتائج، وأخيرا، تتم مقارنتها مع أعمال مماثلة فعلت مؤخرا لإثبات الكفاءة في تنفيذ الخوارزمية المقترحة. يبدأ تشغيل النظام مع مرحلة المعايرة (Calibration Phase)، حيث تم تصميم خوارزمية كشف الوجه للكشف عن وجه المستخدم عن طريق تدريب Support Vector Machine. ثم، يتم استخراج ميزات، وبعد ذلك تتبع للعيون ممكن عن طريق لون الجلد التجزئة أو Skin Colour Segmentation. أجريت زوجان من العمليات الأخرى. النظام العام هو لوحة المفاتيح التي تعمل من خلال حركة العين، من خلال تقنية المسح الضوئي بمساعدة شريك. وقد تم تحقيق مستوى جيد من الدقة، ونفذت عددا من الطرق البديلة ومقارنتها. ويضيف لوحة المفاتيح هذه لمجال البحوث، مع مجموعة جديدة ومبتكرة من التقنيات للكشف عن العين وتتبع. أيضا، لوحة المفاتيح المتقدمة يساعد على سد الفجوة بين الشلل الجسدي وممارسة حياة طبيعية. وهذا النظام يمكن أن يستخدم للمقارنة مع خوارزميات أخرى مقترحة للكشف عن العين، ويمكن أن تستخدم كدليل لكفاءة الجمع بين عدد من التقنيات المختلفة في خوارزمية واحدة. أيضا، فإنه يدعم بقوة فعالية التعلم الآلي (Machine Learning) والخوارزميات القائم على المظهر (Appearance-based Algorithms).

مفاهيم البحث الرئيسية: تتبع العين، كشف العين، تقنية المسح الضوئي بمساعدة شريك، لون الجلد التجزئة، لوحة المفاتيح.

Acknowledgements

I'd like to thank the entire UAEU community for making this achievement possible. Specifically, I thank the EE Department for their guidance and support. My thanks go to my advisor Dr. Qurban Ali for all his assistance and support, and to the program coordinator Dr. Abbas Fardoun and chair Dr. Hassan Noura for all their valuable advices and help, throughout this work. I also extend my gratitude to the IT Department faculty members.

I thank all those who inspire me to move further in my research. I thank all researchers and scholars out there who continue contributing with valuable ideas and inventions. I extend my gratitude to Prince Mohammad bin Fahd University, Al-Khobar, Saudi Arabia.

Last but not the least, I deeply thank my close relatives and friends, who always support me. And I thank all those who willingly participated in the testing phases of this work.

Dedication

To all those who stood beside me throughout this journey and made me who I am today.

Table of Contents

Title.....	i
Declaration of Original Work	ii
Copyright.....	iii
Approval of the Master Thesis.....	iv
Abstract.....	vi
Title and Abstract (in Arabic).....	vii
Acknowledgements.....	viii
Dedication.....	ix
Table of Contents.....	x
List of Tables.....	xii
List of Figures.....	xiv
List of Abbreviations.....	xvi
Chapter 1: Introduction.....	1
1.1 Paralysis.....	1
1.1.1 Problems Faced by Paralysed People.....	3
1.2 Partner-Assisted Scanning Technique.....	5
1.3 Gesture Recognition.....	5
1.3.1 Data Acquisition.....	9
1.3.1.1 Depth Sensors.....	9
1.3.1.2 Image Sensors.....	11
1.3.2 Localisation.....	12
1.3.3 Classification.....	13
1.3.3.1 Gesture Type.....	14
1.3.3.2 Classification Algorithm.....	15
1.3.3.2.1 Artificial Neural Networks.....	15
1.3.3.2.2 Support Vector Machines.....	17
Chapter 2: Methods.....	20
2.1 Eye Calibration.....	20
2.1.1 Calibration Techniques.....	21
2.1.1.1 Method of Capturing Multiple Frames.....	21
2.1.1.2 Method of Bounding Box.....	22
2.1.1.3 Method of Line Laser.....	24
2.1.1.4 Method of Artificial Neural Networks.....	25
2.1.1.5 Method of Support Vector Machines.....	40
2.1.2 Feature Extraction.....	45
2.2 Eye Tracking.....	52
2.2.1 Combination 1: White and Black Pixels in Eyes.....	52
2.2.2 Combination 2: Mathematical Face Measurements.....	54
2.2.3 Combination 3: Eye Detection SVM.....	55
2.3 Classification.....	60

Chapter 3: Results.....	65
3.1 Testing of Calibration Phase.....	65
3.2 Testing of Tracking Phase.....	74
3.3 Testing of System.....	78
Chapter 4: Discussion.....	81
4.1 Calibration Stage.....	81
4.2 Tracking Stage.....	85
Chapter 5: Conclusion.....	89
5.1 Managerial Implications.....	89
5.2 Research Implications.....	91
Bibliography.....	95
List of Publications.....	100
Appendix.....	101

List of Tables

Table 1: Results of training data fitting network (20 hidden neurons).....	32
Table 2: Plotted results of training data fitting network (20 hidden neurons).....	33
Table 3: Measured vs. Desired output values of trained data fitting network (20 hidden neurons).....	34
Table 4: Plotted results of training data fitting network (125 hidden neurons)....	35
Table 5: Results of training data fitting network (125 hidden neurons).....	36
Table 6: Results of training pattern recognition network (20 hidden neurons)....	36
Table 7: Plotted results of training pattern recognition network (20 hidden neurons).....	37
Table 8: Plotted results of training data fitting network (20 hidden neurons), column-wise insertion.....	39
Table 9: Results of trained SVM (linear, 7 SV).....	40
Table 10: Results of trained SVM (linear, 14 SV).....	41
Table 11: Comparing different degree polynomial SVM.....	42
Table 12: Comparing SVM performance with different C values.....	43
Table 13: Comparing SVM performance for different kernels.....	45
Table 14: Different RGB values for eyes with different iris colour.....	47
Table 15: Histogram Equalisation on the different colour channels.....	49
Table 16: Tracking steps in combination 3 (visualised).....	56
Table 17: SVM Eye training results (35-by-70 dimension, 40 training samples).	57
Table 18: SVM Eye training results (45-by-75 dimension, 48 training samples).	57
Table 19: SVM Eye training results (35-by-65 dimension, 52 training samples).	58
Table 20: SVM Eye training results (35-by-65 dimension, 100 training samples).....	58
Table 21: SVM Two-eye training results.....	59
Table 22: Testing results of bounding box eye calibration method.....	66
Table 23: Testing results of data fitting neural network for face detection (20 hidden neurons).....	69
Table 24: Testing results of data fitting neural network for face detection (125 hidden neurons).....	71
Table 25: Testing results of pattern recognition neural network for face detection (20 hidden neurons).....	72
Table 26: Testing results of data fitting neural network for face detection (20 hidden neurons, column-wise).....	72
Table 27: Testing results of tracking combination 1 algorithm.....	75
Table 28: Testing results of tracking combination 2 algorithm.....	77
Table 29: Testing results of tracking combination 3 algorithm.....	77
Table 30: Testing results of integrated system.....	78
Table 31: Pros and cons of the different calibration methods.....	82
Table 32: Comparison between the different calibration methods.....	84
Table 33: Pros and cons of the different tracking methods.....	85

Table 34: Failure of eye colour segmentation in eye tracking.....	86
Table 35: Comparison with other research works.....	92

List of Figures

Figure 1: Causes of Paralysis in the U.S.....	2
Figure 2: Different Types of Paralysis.....	4
Figure 3: 3D model-based algorithm for Gesture Recognition.....	7
Figure 4: Skeleton-based algorithm for Gesture Recognition.....	7
Figure 5: Appearance-based algorithm for Gesture Recognition.....	8
Figure 6: Kinect Sensor, RGB camera and depth camera.....	10
Figure 7: Bright/ Dark Pupil Effect.....	14
Figure 8: Artificial Neuron.....	16
Figure 9: SVM margin.....	17
Figure 10: Calibration by multiple frames.....	22
Figure 11: Multiple captured frames after processing (frames 2-15 from top right).....	22
Figure 12: Bounding Box calibration.....	24
Figure 13: Bounding Box implementation.....	24
Figure 14: Line Laser implementation.....	25
Figure 15: Brightest laser reflection in green box is detected as the user's face.	25
Figure 16: First row left-right: grey scale image, AHE on RGB channels, AHE on R channel; second row left-right: AHE on B channel, AHE on G channel, HE on RGB channels; third row left-right: HE on RGB channels individually.....	27
Figure 17: Image shape not distorted after rescaling to 25-by-25 due to cropping (left); image shape distorted after rescaling to 25-by-25 when cropping is not done since original images captured by webcam are not square (right).....	27
Figure 18: Steps of processing images prior to insertion into network as training input.....	28
Figure 19: Displaying how image pixels inserted row-by-row as input into network.....	28
Figure 20: Total 625 (25 by 25) pixel values of processed grey scale image.....	28
Figure 21: A section of the training input file.....	29
Figure 22: Input image displaying grey scale pixel values inserted into input file.....	30
Figure 23: A section of the output file in training.....	31
Figure 24: A rough diagram of the neural network to be trained.....	32
Figure 25: Network hidden layer, 20 neurons (left); network output layer, 4 neurons (right).....	32
Figure 26: Inserting pixel values column-wise.....	38
Figure 27: Plotted SVM weight matrix.....	43
Figure 28: Plotted weight matrix for different C values.....	44
Figure 29: Eye detected in b/w image.....	45

Figure 30: Extracting eye measurements.....	46
Figure 31: Human eye.....	48
Figure 32: Brightest pixels in yellow (cornea), darkest pixels in red (pupil).....	50
Figure 33: Extraction of RGB values of selected skin area.....	51
Figure 34: Face detection by skin colour segmentation.....	51
Figure 35: Extracting face measurements.....	52
Figure 36: Steps of tracking combination 1.....	52
Figure 37: Dark pixels detected.....	53
Figure 38: Bright pixels detected.....	53
Figure 39: Calculating centre of both dark and bright pixels.....	54
Figure 40: Detected Eyes Centres.....	54
Figure 41: Steps of tracking combination 2.....	55
Figure 42: Steps of tracking combination 3.....	55
Figure 43: Eye search area (face golden ratio).....	60
Figure 44: Detection of eye ball position during classification stage.....	61
Figure 45: Eye indicating pressing key (right); eye indicating no key to press (left).....	61
Figure 46: Designed PAS keyboard.....	61
Figure 47: Communication between the three eye tracking system files.....	63
Figure 48: Flow chart of PAS eye tracking system.....	64
Figure 49: Frame 7 detects wrong pixels as eyes in multiple frames method....	65
Figure 50: Participant eyes detected by bounding box method.....	66
Figure 51: Steps of face detection by SVM.....	73
Figure 52: Face detected in yellow box by trained SVM.....	74
Figure 53: Middle of box of SVM detected face matches exactly the face centre.....	74
Figure 54: Comparison of Accuracy Rate of the different tracking algorithms...	87
Figure 55: Comparison of Speed of the different tracking algorithms.....	87
Figure 56: Head Tilt Graph along the XYZ axis.....	93

List of Abbreviations

CAMSHIFT	Continuously Adaptive Mean Shift
CCD	Charge-Coupled Device
GELM	Graph Regularised Extreme Learning Machine
HSI	Hue, Saturation and Intensity
HSV	Hue, Saturation and Value
LAB	L Lightness, a,b colour-opponent dimensions
MCT	Modified Census Transform
OpenCV	Open Source Computer Vision
OpenNI	Open Source Natural Interaction
PAS	Partner Assisted Scanning
QVGA	Quarter Video Graphics Array
RANSAC	Random Sample Consensus
RGB	Red Green Blue
SDK	Software Development Kit
SIFT	Scale-Invariant Feature Transform
SV	Support Vector
TLD	Tracking Learning Detection
VGA	Video Graphics Array

Chapter 1: Introduction

1.1 Paralysis

Abdullah Bani'mah was a healthy young man, until one day, at the age of 18, a jump into a shallow pool at a Jeddah swimming club have changed his life completely. He stayed inside the pool for 15 minutes, until his friends rescued him. He didn't die, but his brain has been severely damaged due to the long stay in water, which ultimately led to the paralysis of his entire body. Now, Bani'mah can only control his head while his body has been paralysed for his whole life. Such is the case of many people who met with accidents that left them paralysed for their entire lives. Unlike many other disabilities, Paralysis is special in its kind, since any normal person could find himself/herself suddenly paralysed for the entire life due to an unexpected accident. There are no accurate studies of the total number of paralysed people worldwide, especially due to the fact that this number keeps increasing every year due to many reasons like accidents, wars, diseases and so on.

The Kingdom of Saudi Arabia, KSA, is the biggest country in the Middle East with a population of around 28 million according to the 2010 census. Out of this nearly 30 million, an estimated 4% are disabled (Al-Gain, 2002). Of those disabled, physical disability tops to number one, mainly due to road accidents. Not much research has been conducted on the exact number of physically disabled people in Saudi Arabia. However, one thing for sure which any person living in Saudi Arabia knows, is that road accidents are frequent. According to the figures by the Saudi health ministry, 598,300 accidents have occurred in 2012, which accounts to an average of 1,614 a day and of 67 an hour (Al-Jadid, 2013).

According to a study done by the Christopher & Dana Reeve Foundation (2002), one person in every 50 suffers from paralysis in the United States (U.S.); which is around 6 million people (around 2% of the U.S. population) And this number is higher than their previous estimates by 40% (Reeve, 2002). Figure 1 shows how stroke was the cause for majority of the paralysed in the U.S. Other two major causes were spinal cord injury and multiple sclerosis.

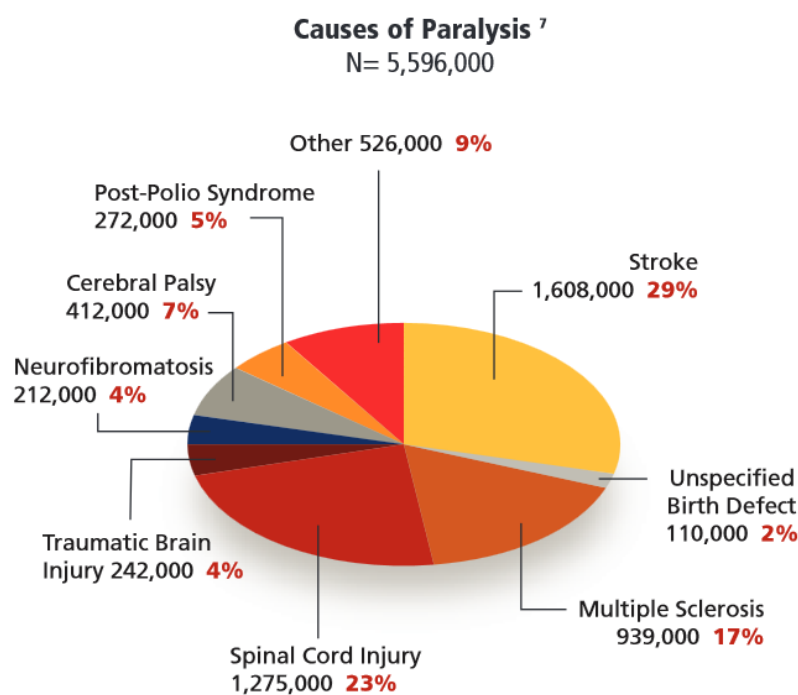


Figure 1: Causes of Paralysis in the U.S (Reeve, 2002)

Paralysis, according to the National Health Service (2014) in the United Kingdom, is loss of the ability to move one or more muscles. It may be associated with loss of feeling and other bodily functions. It can be seen as a form of nerve damage, due to the inability of the brain to control muscles due to damage in the nerves of spinal cord.

1.1.1 Problems Faced by Paralysed People

Despite the large number of disabled people in the Middle East (for instance, around 1 million in KSA) and worldwide, they face many difficulties in the society. First is the lack of research to know the exact number of disabled people and the types of disabilities they face (Al-Gain, 2002). There is no specialised institute to collect data such as those existing in the U.S. This leads to some level of negligence to the needs and requirements of this important group. As stated by Al-Gain, disability is regarded as shameful in some families, and not all families admit that their relative is disabled. Second, the facilities in public do not confine to the needs of the physically disabled. A simple example is the need for building of ramps in all places to help people in wheelchairs. Thirdly, rehabilitation centres, both public and private, do exist and provide care and treatment to physically disabled people (Al-Gain, 2002). However, there is a need for more such centres, due to the fact that the number of physically disabled people increases annually from road and related accidents. Other issues are also faced by physically disabled people in Saudi Arabia. The quality of life of people (majority, young adults) with spinal cord injuries due to road accidents is greatly affected due to factors like accessibility, financial status and employment (Al-Gain, 2002). In terms of education, there are special schools and government owned Social Rehabilitation Centres and institutions for the children with special needs, including physical disability. For instance, the Augmentative and Alternative Communication (AAC) is being used as means of communication with children with difficulty in expressing themselves due to physical disability or other reasons like deafness. Although an effort is being made to provide good quality education services in these places, more is needed to reach high standards in the

provision of educational services. Thus, the lack of research in the field of disability in KSA leads to ignorance towards the needs and requirements of the disabled in the country. While there has been an improvement in the quality of life of the disabled (around 1 million) in the country, more needs to be done to help people physically challenged lead a normal life and become part of the society.

People living with paralysis face difficulties in their lives. However, the level of difficulties varies, since paralysis happens differently from one person to another depending on the cause of paralysis. A severe case of Paralysis is Tetraplegia/Quadriplegia where both the arms and legs are paralysed (National Health System, 2014), as in Figure 2. Thus, here, a person needs help from a caregiver the whole time. Basic life activities become difficult or impossible to do.

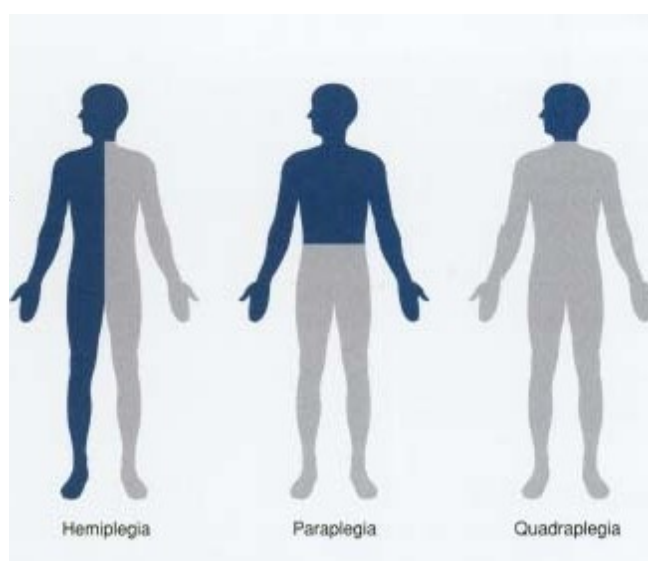


Figure 2: Different Types of Paralysis (Humanillness.com)

As such, paralysed people need to cope with their new life and overcome the difficulties. The role of the society is to help paralysed people cope with their lives and develop ways for them to easily perform basic life activities.

1.2 Partner-Assisted Scanning Technique

According to the Cincinnati Children's Hospital Medical Centre (December 2011), Partner-Assisted Scanning is a way of communication for people with severe motor problems or visual problems or cannot speak, yet they can think and possess good mental abilities. In this way of communication, the care giver presents a set of words or letters to the disabled patient, and the patient can select from among those letters or words in order to express his/her thoughts. Selection might happen by pointing or blinking of the eye depending on the capabilities of the disabled person. This method is used to help children as well as adults who suffer immobility due to late stages of ALS, multiple sclerosis, or severe injuries. In ALS, eye movement is usually unaffected until later stages unlike other body parts. This is because of the difference between the extra ocular muscles of the eyes and the skeletal muscles of the rest body parts.

A famous story of the effectiveness of partner-assisted scanning technique is that of Jean-Dominique Bauby. Bauby was a French journalist who suffered from a stroke at the age of 43 that left his body completely paralysed, a condition called locked-in syndrome. He could only move his left eyelid. During this period of entire paralysis, Bauby wrote an entire book, *The Diving Bell and the Butterfly*, with the help of another person (a helper) through partner-assisted scanning, by moving only his left eyelid. The movements of his left eyelids indicated which word he wanted his helper to write down.

1.3 Gesture Recognition

Gesture Recognition is the interpretation of gestures of a human by means of mathematical algorithms. The gestures can be formed by face, hand or any part of the

human body. Gesture Recognition is mainly useful in providing an easier and more natural way to interact with devices in comparison to the traditional input devices like mouse and keyboard. It can also prove useful to physically disabled people when interacting with devices. Some technologies developed that proved the possibility of gesture recognition to be implemented in input devices is the Sixth Sense Technology (Mistry & Maes, 2009). This is done in such an intelligent manner that even slightest gestures by a user can be interpreted by the Sixth Sense device and the actions (like taking a picture) are made. The main idea behind this technology is to change the way people interact with devices and bridge the gap between the physical and digital world, as stated by Mistry (2009). Techniques that help in gesture recognition include computer vision and image processing.

Gesture Recognition is a hot research topic today, and new algorithms are being invented. No algorithm has proved to be the best until now, but there are many inventions that came with good performance that proves a successfully working algorithm. An example of such an invention is that of the Kinect Sensor developed by Microsoft, and the Wii Remote developed by Nintendo. Gesture Recognition requires the use of input devices. A large variety of input devices exist based on the application, algorithm used and other factors. First input devices to be implemented were data gloves, where every movement of the hand including the fingers produced signs that could be detected. Other input devices are cameras, like the depth cameras, stereo cameras and 2D cameras.

In terms of the algorithm used for gesture recognition, there could be two main types (Mitra & Acharya, 2007). One is 3D model-based algorithm (Pavlovic et al. 1997). Here, the body parts are captured in 3D, from which relevant information

are extracted for gesture recognition. This approach is common in the animation industry. The second approach is Appearance-based algorithm (Pavlovic et al. 1997). This approach is a simple one that depends on the images or video captured directly by camera in order to interpret the body part gesture. The former approach is more complicated in comparison to the latter, and yet to be developed further. The template-based are part of this approach (Pavlovic et al. 1997), where sets of points on the outline of the body part (mostly, hand) are used for outline approximation to detect gesture. Another algorithm used in gesture recognition is the Skeletal-based (Pavlovic et al. 1997). In this approach, only key parameters like joint angles and segment lengths are analysed in contrast to the 3D model-based algorithm where the entire 3D picture is analysed. This is a simple and fast approach, and can be categorised under the 3D model-based algorithm.

The above discussed algorithms can be illustrated in the Figure 3, Figure 4 and Figure 5.

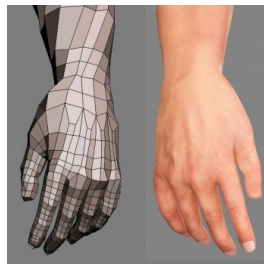


Figure 3: 3D model-based algorithm for Gesture Recognition (Creativecrash.com, 2010)

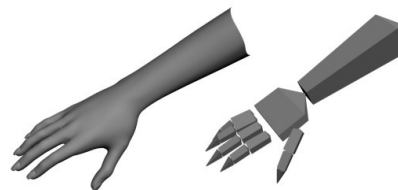


Figure 4: Skeleton-based algorithm for Gesture Recognition (Razvan, 2011)



Figure 5: Appearance-based algorithm for Gesture Recognition (Ramirez-Cortes, Gomez-Gil, Sanchez-Perez & Prieto-Castro, 2009)

Under each type of algorithm, certain problems are faced. For instance, under the Appearance-based approach (Pavlovic et al. 1997), problems related to the surrounding environment are mostly common. These include image/video capturing under inconsistent lighting conditions in different environments, or analysing images with colourful or moving background of the surrounding environment. Thus, the challenge in gesture recognition is to choose the right algorithm and input method that can ensure the best results. In all approaches, gesture recognition comprises three main steps, namely: Image Acquisition, Localisation (that can be further divided into two phases: Segmentation and Tracking), and Gesture Classification. These are discussed in the coming sections.

Eye Tracking is part of gesture recognition. This conclusion is justifiable since facial movements (under which eyes are categorised) are one of the types of gestures identified under gesture recognition. Hence, eye movement is considered a gesture. Eye tracking can refer to the tracking of the position of the eye or tracking of the movements of the eye. Three main types of eye tracking exist, depending on the measurement technique used: 1) First is attaching contact lenses or similar objects to the eye that helps measure the accurate eye dynamics and movements, 2) Second is to direct an IR light source onto the eye and capture information of the eye movements through video cameras (also known as video-oculography), information about the pupil and iris are gathered here, 3) And third is to place electrodes around

the eyes and measure the electric potentials (also known as Electrooculogram (EOG)), thus tracking the eye movement even when the eyelids are closed. Two common eye movements usually tracked are saccades (fast rapid movement) and fixations (eyes fixed at one point). Each of the gesture recognition steps are discussed below:

1.3.1 Data Acquisition

The first step in achieving any gesture recognition algorithm is data acquisition. Data needs to be acquired before being processed for recognising gestures. This data could be an image or video. Even for image, there exist a couple of types of images captured, depending on the capturing device used, the method chosen to acquire the data, and how the gesture recognition algorithm works. Sensors play a major role in data acquisition. Sensors could be wearable sensors like data gloves, or external sensors like video cameras. An example of the latter is the CMOS (Complementary Metal-Oxide Semiconductor) sensor found in cameras. Video cameras have become more common in data acquisition for gesture recognition in comparison to data gloves. This is because the data gloves approach does not sound so practical since users need to wear gloves, as well as they are more expensive. In terms of eye tracking, data acquisition can be accomplished through capturing images or video of the eye by cameras, or through attachments placed on the eye (like eye contact lenses), or by placing electrodes around the eyes.

1.3.1.1 Depth Sensors

In terms of cameras, certain problems are faced by the 2D or video cameras due to a couple of reasons like occlusions, lighting changes, rapid motion, or other

skin-coloured objects in the surrounding background (Suarez & Murphy, 2012). These problems are not found in depth cameras or 3D cameras and stereo cameras. However, this does not imply that the latter cameras are better than the former ones. It is a matter of choice depending on the approach used. But there is no doubt that the invention of the Kinect sensor has created a revolution in gesture recognition and computer vision. Very similar to it is the ASUS Xtion Pro. Many researchers have used the Kinect sensor to build algorithms for gesture recognition. Microsoft's Kinect consists of a QVGA (320×240) depth camera and a VGA (640×480) video camera (Suarez & Murphy, 2012), as shown in Figure 6. The depth camera works on the principle of light. An infrared (IR) emitter projects a sequence of dots in the front of the IR camera. This IR camera can determine the distances of objects based on the distance between the scattered dots. In addition, there are also other alternatives like the open source OpenNI and closed source NITE middleware for OpenNI, that help in gesture tracking similar to Kinect.

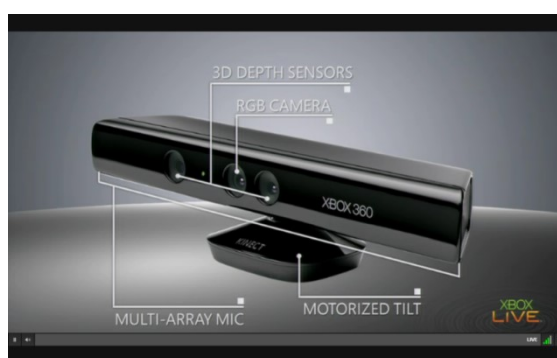


Figure 6: Kinect Sensor, RGB camera and depth camera (Shawn, 2010)

As discussed in (Suarez & Murphy, 2012), other depth sensors include Time of Flight (ToF) cameras and stereoscopic cameras. Example of the former is Mesa Imaging's SwissRanger, which works by measuring round-trip flight-time projected and reflected back or measuring the phase-shift of reflected light. Example of

stereoscopic cameras is Point Grey's Bumblebee. These cameras work by capturing two simultaneous images and using registration methods in disparity maps for approximation of per-pixel depth. Other methods for depth image acquisition also exist. One is to project different thickness light bands onto objects for depth calculation. Such a method depends on light or shadow differences for depth estimation.

1.3.1.2 Image Sensors

Image sensors are the video or digital cameras that are more common than depth cameras. Two types of image sensors are the CCD (semiconductor charge-coupled devices) and the active pixel sensors in CMOS technology (Litwiller, 2001). They mainly work by converting light into electrical signals. The two types function in the same way, as is explained in (Litwiller, 2001). In CCD, when light falls on the chip, the charges at each photo sensors are converted one pixel at a time to a voltage. The CMOS contains active pixel sensors where light is converted at the rate of one row of pixels at a time. The active pixel sensors are commonly used in webcams and phone cameras. The advantage of CMOS over CCD is that CMOS sensors are cheaper, and hence more preferred in manufacturing. Image sensor cameras may also contain colour image sensors that help separate each of the RGB channels. Example of such colour image sensors are Bayer filter sensor, Foveon X3 sensor and the 3CCD. Each of these sensors uses a different mechanism for colour separation.

In the project developed by Mori et al. (2010), a stereo motion capturing system is used for data acquisition. Akl et al. (2011) uses WiiRemote, while Ren et al (2013) and Celebi et al. (2013) implement data acquisition using Kinect sensor. Frolova et al (2013) implements data acquisition through PrimeSense 3-D camera.

Two types of sensors, inertial and optical, are implemented by Zhou et al (2014) and Chen et al. (2013), but the latter also uses the WiiRemote to measure acceleration and angular speed of body part. Nguyen et al. (2013) and Arora et al. (2014) use webcams for data acquisition, while Alon et al. (2009) uses Windows PC camera itself to capture gesture images.

1.3.2 Localisation

Localisation is an important step in gesture recognition, which constitutes two steps: Segmentation and Tracking. Segmentation is the problem of determining which pixels in an image constitute the hand (Suarez & Murphy, 2012) or the face. A variety of techniques can be implemented in the segmentation stage like Depth thresholding, Viola Jones Object-Detection Method, Karhunen-Loeve Decomposition (Kirby and Sirovich, 1990), skin-colour mapping, Cascaded Classifiers on Haar-like features (Lienhart and Maydt, 2002), clustering, region growing, static background subtraction, shadow analysis. In the Sixth Sense Technology (Mistry & Maes, 2009), the user wears colour markers around his/her fingers to simplify detection of fingers' movements by the camera. Tracking is determining the trajectory of the desired object in the sequence of images (Suarez & Murphy, 2012). There are different methods implemented like OpenNI through NITE middleware and Microsoft Kinect SDK (Ren et al., Celebi et al., 2013), Kalman filter (Esme, 2009) as shown in equation in (1), mean shift and CAMSHIFT.

$$X_k = K_k \cdot Z_k + (1 - K_k) \cdot X_{k-1} \quad (1)$$

A number of different approaches have been implemented for eye tracking. Some projects apply face detection prior to eye detection. Praglin and Tan (2014) perform eye detection by colour methods, after which they mask out non-face regions until only two remain where they compute the centroid of connected regions. After eye detection face is detected by computing mean colour mark pixels in the colour space of the mean colour in the image. Gaze estimation is performed using SIFT along with RANSAC and homography model. Liu et al (2010) apply the famous Viola-Jones classifiers for face detection, along with a mean shift tracking algorithm. Zia et al. (April 2014) performed skin colour segmentation for face detection, followed by circular Hough Transform to detect the iris. Face detection has been achieved by Adaboost classifiers using MCT features in the project of Choi et al. (July 2011). Both Bengoechea et al. (September 2012) and Fernandez et al. (April 2014) implement Viola-Jones algorithm for face detection. The former combines it with TLD algorithm and Lucas-Kanade Algorithm, while the latter implements along ANN. A number of researchers prefer combining methods together, as is the project by Majumder et al. (March 2011). Skin colour segmentation is performed in the HSL colour space face detection in the paper by Fosalau et al. (August 2011). Machine learning was performed by Jiao et al (July 2014) on SVM, GELM and KNN. Template matching and adaptive block matching search techniques were implemented by Abdel-Kader et al (2014).

1.3.3 Classification

After the hand has been segmented and tracked, the hand actions detected need to be classified into meaningful gestures. Before classifying the hand actions,

the gesture type needs to be defined. After the gesture type is determined, the hand trajectory is fed as input into a classification algorithm for hand gesture recognition.

1.3.3.1 Gesture Type

As explained by Suarez and Murphy (2012), four categories of gestures are defined: (i) gesticulations (speech used for emphasis), (ii) emblems (gesture codes without sound), (iii) pantomimes (not part of any code, no speech), and (iv) sign language (replacing speech). Different gesture types exist depending on the type of application for which hand gesture recognition is performed. For instance, in sign language interpretation, the gesture type will be the dictionary for the ARSL (Arabic Sign Language), BSL (British Sign Language), ASL (American Sign Language), or the sign language of any other county. An example is the projects by Nguyen et al. (2013) and Alon et al. (2009). Some define their own gesture type, like Ren et al. (2013).

In terms of eye tracking, many eye gestures can be tracked like eye movement tracking, eye gaze estimation, or eye blink detection. Eye gaze estimation can be achieved by observing the corneal reflections on the cornea due to an IR light source. In this term, two effects are common, the bright pupil and dark pupil respectively (Figure 7).

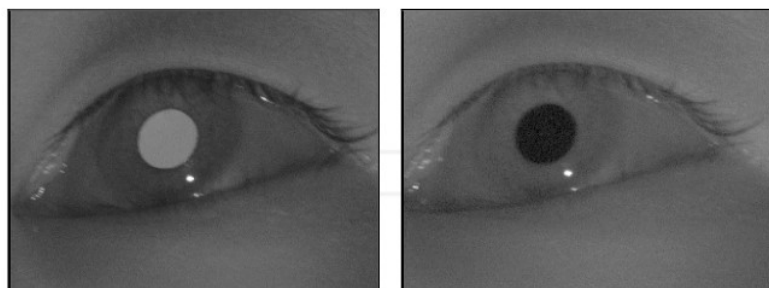


Figure 7: Bright/ Dark Pupil Effect (Meunier et al., 2009)

The former is when the light source is within the inclination path of the eye, thus causing the light reflection from the pupil to fall back into the camera. The latter is when the light source is away from the inclination path of the eyes, and the light reflected from the pupil goes away from the camera in a different direction. In terms of eye movements, saccades and fixations are the two well known. Saccades are the fast movements of the pupil, while fixations are when eyes focus on an object of interest.

1.3.3.2 Classification Algorithm

Different classification algorithms exist, and none has proved to be the best one. Each algorithm has its own pros and cons; however certain algorithms have been more implemented than others. Choosing one might depend on the gesture type extracted. Some of the classification algorithms are Hidden Markov Models (HMM), k-Nearest Neighbours (k-NN), Neural Networks, Support Vector Machines and Template Matching. These methods have been implemented in papers reviewed by Suarez and Murphy (2012).

1.3.3.2.1 Artificial Neural Networks

Any complex task, when broken down into simpler elements, becomes easy to solve. A network applies the concept of 'divide and conquer' to solve complex tasks (Gershenson, 2003). A network consists of (i) a set of nodes and (ii) connections between those nodes. The nodes are seen as computational units which receive an input, process it, and produce an output. The connections represent the flow of information between the different nodes. Overall, the interactions inside a network led to certain behaviour. Example of a network is the Artificial Neural Networks (ANN). ANN is a concept inspired from the central nervous system in a

human body that consists of neurons in a similar structure to a network. ANN helps in machine learning and pattern recognition. They work in the same way a human neuron works. Inputs are multiplied by weights and computed by a mathematical function to determine the neuron's activation, and another function computes the output of the neuron, as in Figure 8 (Gershenson, 2003). Weights can be adjusted to obtain the wanted output for specific inputs. The weights are usually adjusted by an algorithm by a process called learning or training. Many ANN have been developed for different applications. In each network, certain features might be different like functions, accepted values, topology or learning algorithms.

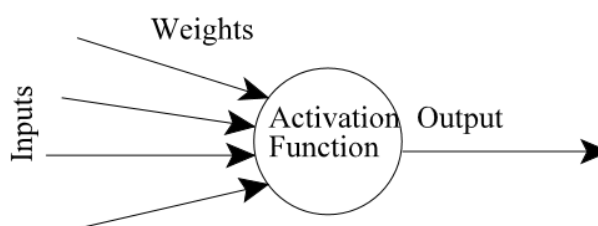


Figure 8: Artificial Neuron (Gershenson, 2003)

Many algorithms have been implemented in ANN. One of the most common is Backpropagation algorithm. This is used in layered feed-forward ANNs, which means those neurons are organised in layers and the signals are sent forward and the errors are propagated backwards (Gershenson, 2003). There is the input layer, the output layer, and some intermediate hidden layers. This algorithm is a supervised one in which examples of the inputs and outputs to be computed are provided to the network and the error is calculated. The error is the difference between the actual and expected results. This error is to be reduced, until the training is learnt by the ANN. The activation function in Figure 8 in the Backpropagation algorithm is set in relation to a weighted sum of inputs multiplied by their weights. The training process aims at obtaining a desired output when certain inputs are given. More details are

given in Gershenson's paper (2003). Because of the way they work, neural networks are well suited to be used in classification problems in designing gesture recognition (Stergiopoulou and Papamarkos, 2009), air-traffic landing sequences (Memon, 2008), Clustering of Wear Particle Measurements (Memon, 2007; Memon, 2006), Crime Investigation, and Analysis (Memon, 2003) etc.

1.3.3.2 Support Vector Machines

Support vector machines (SVM) have produced good results in pattern recognition and classification. SVMs are implemented when data is classified into two classes. It searches for a best hyperplane to separate the two classes apart, as mentioned by Hearst et al. (1998). Searching for the best hyperplane means searching for the largest margin (that with maximum width) that separates all data points of one class from all data points of the other class. This is illustrated in the figure below. Support vectors are the points closest to the separating hyperplane. The '+' points belong to one class, and the '-' points belong to the other class respectively. As an illustration, this is shown in Figure 9.

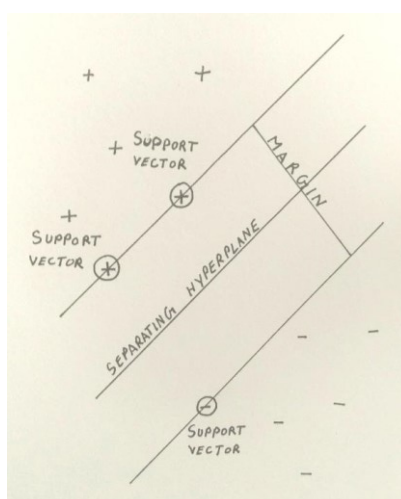


Figure 9: SVM margin

Mathematically, SVMs are implemented by a set of sequential equations. The separating hyperplane is given by the following equation:

$$\langle w, x \rangle + b = 0, \quad (2)$$

where $\langle w, x \rangle$ is a dot product of weights and data points, b is bias

To achieve the best separating hyperplane, the following equation needs to be satisfied:

$$y_i(\langle w, x_i \rangle + b) \geq 1 \quad (3)$$

where y is +/- one depending on the class

To simplify quadratic equations, Lagrange multipliers α_i are applied with the constraint and subtracted from the condition for maximal margin, which is

$\frac{1}{2} \langle W, W \rangle$. Thus, after applying primal Lagrange multipliers,

$$L_P = \frac{1}{2} \langle W, W \rangle - \sum_i \alpha_i (y_i(\langle w, x_i \rangle + b) - 1) \quad (4)$$

After equating L_P to zero, and solving we get:

$$w = \sum_i \alpha_i y_i x_i \quad \text{and} \quad 0 = \sum_i \alpha_i y_i \quad (5)$$

Thus the dual Lagrange multipliers become:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (6)$$

Values of α_i lies in the range $0 \leq \alpha_i \leq C$, where C is a constraint which keeps values of α_i within a limited range.

On the other hand, for nonlinear transformations, Kernels are often applied since a simple hyperplane cannot separate the two classes of data points apart. The concept

of the Kernel method is that there exist a function φ that maps x to a linear space S , such that

$$K(x,y) = \langle \varphi(x), \varphi(y) \rangle \quad (7)$$

This dot product takes place in the linear space S .

A number of issues are faced by physically challenged people in the Middle East region. Due to lack of research, there is negligence to some basic rights and requirements of such category of people in the society. There is a lack of facilities that help ease their life. As such, they miss on a lot of employment and other opportunities. Surely, advancements in technology can help improve the life of physically disabled people. Gesture and face recognition can serve as an alternative to people with paralysis. With regard to gesture and face recognition, a number of methods have been proposed in the literature and discussed briefly in the previous sections. The issue is that, with different proposed algorithms, no method has proved to be the best in terms of detection. Each method, whether colour segmentation or template matching or machine learning, has its pros and cons. Common limitations exist in many of the discussed papers, like inability to detect the face when oriented differently or the surrounding background is moving. Also, in some cases combination of different methods for face detection produces good results. Thus, face detection algorithms are still a topic of research to come up with better results than the existing ones. Many challenges exist in face and eyes detection and tracking like lighting conditions. These challenges are considered when detection and tracking methods are chosen.

Chapter 2: Methods

Gesture Recognition can be utilised to serve people who need it the most, the paralysed who face difficulties coping with the life around them. Studying the different approaches in gesture recognition can help build a system that serves those people. Building an eye tracking keyboard can solve many difficulties faced by people with physical difficulties. Also, it transforms partner-assisted scanning technique to a new level, from implementation on papers to electronics.

Building an eye movement tracking system, as discussed in chapter 1, involves three main stages: Eye Detection or Calibration, Tracking and Classification.

2.1 Eye Calibration

Any device that tracks an object (here, eyes) will first requires a calibration step where the system is able to recognise the object to track. A good example is smart board which displays dots at the start to recognise the user's smart pen, calibrate with it and be able to track its movements later on. Another good example is the fingerprint password access feature added to the iPhone devices that requires calibrating the user's fingerprints at the beginning before using that password feature. Devices designed for users with disabilities are usually used by a single person, unlike those designed for public use like the computers in universities and other public places. As such, calibration is required only once. In this work, the calibration phase is aimed to be once when the user uses the system for the first time. After the calibration phase of eye detection of the user, the system can be easily used by that user skipping the calibration phase.

A couple of methods were explored to come up with the most efficient and with best results. The main goal behind calibration stage is to detect the eyes and extract features that will help in eye tracking later on. Here, focus was building a quick and not time-consuming calibration algorithm. The input device is a Logitech Webcam C110, with an in-built CMOS sensor and a noise reduction quality (a method similar to video-oculography). This camera takes pictures at 30 frames per seconds at a resolution of 1.3 Mega pixels. It is an inexpensive and affordable camera. The algorithms designed are based on Appearance-based work on 2D images captured by the camera.

2.1.1 Calibration Techniques

2.1.1.1 Method of Capturing Multiple Frames

Eye detection happens directly without the need for face detection. A person blinks her/his eyes naturally. This quick movement of the eyes is utilized to detect the user, assuming that the background is stationary. Since the system is designed for indoor use and specifically for users with physical difficulties, the surrounding environment will most probably be stationary. Even if there is a movement, the system focuses on the movements in the middle centre of the screen (where the user is most probably seated).

This method is software implemented, where the user is made to stare at the screen and blink naturally for around 0.5s, during which multiple frames are captured by the camera. By comparing multiple captured frames, the system will be able to detect the eyes of the user which are under movement with the help of simple image processing techniques. As seen in Figure 10, this method mainly involves subtraction

between frames, processing the frames and selection of the best subtraction results. Best subtraction results mean that the subtraction took place between one completely closed eye frame and one completely opened eye frame. As such, the subtraction results show the eyes shape and size correctly. The final result is a black and white image with the subtraction results (eyes) in white pixels and the remaining picture in black pixels. Figure 11 shows sample images after processing and how the best frame is selected. The best frame selected is frame 10 with maximum white pixels representing the eyes accurately.

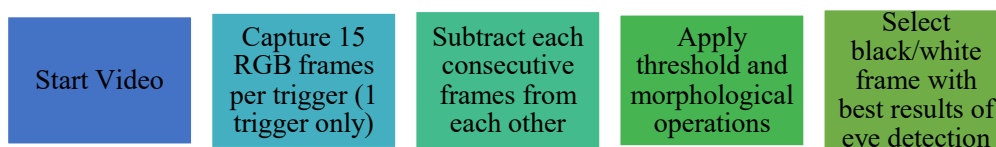


Figure 10: Calibration by multiple frames



Figure 11: Multiple captured frames after processing (frames 2-15 from top right)

2.1.1.2 Method of Bounding Box

This approach also detects the eyes directly without the need to detect the face prior to eye detection. The aim of the calibration is to detect the user's eyes accurately and locate its position. As such, in this approach, bounding box visually appears on the system screen in front of the user. The user is required to position

him/her in such a way that the two eyes are positioned inside this box. Thus, this approach is named 'bounding' box. This method is very simple and works efficiently. The user is only required to follow two instructions: 'open your eyes' and 'close your eyes', while positioning the eyes inside the bounding box that appears on the system screen in front of her/him. Since the position and size of the bounding box is already known, the position of the user's eyes can be known and extracted precisely. This method is flexible and the size of the bounding box can be chosen such that it allows the user to use the system from a range of distances measured from the system screen. This method is similar to the multiple dots calibration of smart boards, in which the position of the dots are previously studied and known. A pause of 5 seconds is added before each frame is captured to allow user to adjust face position with respect to the bounding box and respond at ease to the sound file played before frame capturing instructing the user to open and close eyes.

The dimensions of the bounding box have been decided to be of width 213 pixels and height 63. This size of the box has been chosen by trial-and-error method in such a way that will provide freedom to the user to be seated in any distance from 23.75 inches (60 cm) to 71.25 inches (180 cm) from the computer screen. A distance of less than 23.75 inches might be so close that the eyes fall outside the bounding box, while beyond 71.25 inches from the computer screen might be so far in such a way that the closing and opening of the eyes are not properly detected by the program. Both cases might lead to improper detection of the eyes. According to Apple Inc. (2015), a distance of 18 – 24 inches is the comfortable zone to place the computer screen away from the eyes. This means that a distance of 23.75 – 71.25 inches is a good distance to avoid eye discomfort of the user. Figure 12 shows the

steps in calibration by the bounding box method and Figure 13 shows the implementation of the bounding box method.

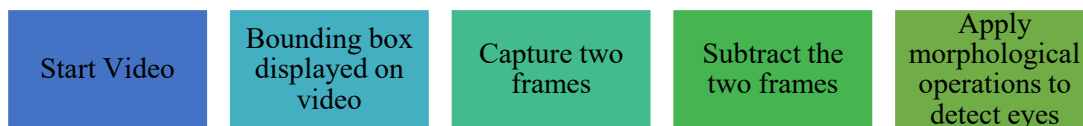


Figure 12: Bounding Box calibration

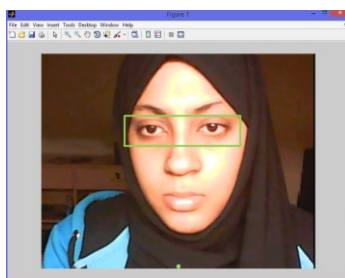


Figure 13: Bounding Box implementation

2.1.1.3 Method of Line Laser

This method involves face detection prior to eye detection. Figure 14 shows the steps of implementation of line laser calibration method. In this method, hardware setup is required. Laser has been implemented in a number of systems because of its unique features, one of which is its brightness on reflection. This feature is exploited in laser keyboards, as in (Zalkassim, 2012). In laser keyboards, the reflection of the line laser onto the typing finger is captured by the camera and indicates that a key has been pressed. Similarly, in this calibration method, a line laser (a point laser can be an alternative) helps in detection of the user in front of the system. This method depends on the fact that a user is the nearest object to the system screen. Thus, a line laser is powered on a board and placed under webcam at a horizontal position to the user. In this set up, it is projected at the lower part of the user's face. For calibration, the laser is switched on and calibration starts. The laser

line spans horizontally all objects in front of the screen. Since the user is the closest object to the screen, the reflection of the laser on the user's face is the brightest. Thus, the brightest reflection is recognised by the camera. For this method, only one image is taken by the camera. A few processing steps are done, and the region on which the laser reflection is brightest is detected as the face. As in Figure 15, only one frame capturing is required.

This method is similar to physical methods implemented for eye and gesture detection. This is because the main working of this technique is physical (through the use of laser). Many physical approaches have proved successful in human computer interaction like the use of electrodes for detecting eye movements, the use of hand gloves to detect hand movements and the use of coloured markers on hand fingers in the Sixth Sense Technology (Mistry & Maes, 2009).



Figure 14: Line Laser implementation



Figure 15: Brightest laser reflection in green box is detected as the user's face

2.1.1.4 Method of Artificial Neural Networks

Research examples of neural network implementation for the purpose of camera calibration can be found in literature (Memon and Khan, 1998; Memon and Khan 2001). There are a number of different neural networks that were trained. The

performance of neural networks depends largely on the training input image samples, as well as on other factors like the number of hidden layers chosen. A neural network is trained to detect a face based on common dark spots in any human's face. But before training, both the training input file and desired output file needs to be prepared. First, images of the people's faces (training samples) are captured in RGB form at a resolution of 640-by-480 pixels. This is the standard resolution of the webcam used and is same as the resolution of pictures captured during the tracking process. Next step is to convert the RGB images into grey scale. This is done since the three colour channels are not required for training the network. Since the network will detect faces by recognising dark places in the face, images will be inserted as input to the network in the form of grey scale values. Third step is to increase the contrast of the sample face images by adaptive histogram equalisation of all channels. Two contrast techniques were tried. One is histogram equalisation (HE) of each colour channel and all channels. Second trial was adaptive histogram equalisation (AHE) of each colour channels and all channels. While some produced little contrast, others produced too much contrast. Adaptive histogram equalisation of all three channels at once produced the most desirable contrast between dark and bright areas in the face (not too much and not negligible contrast). Overall, adaptive histogram equalisation produces better results, as shown in Figure 16.

Last step in processing images before inserting them as training input into the neural networks is to resize images. Images captured at 640-by-480 pixels resolution are too big to be inserted as input. Since the interest is to detect dark and bright areas in the faces, resizing will not affect the images. The images are first cropped to a size of 368-by-336. This cropping is not necessary, but has been done only for neatness and simplicity purposes.



Figure 16: First row left-right: grey scale image, AHE on RGB channels, AHE on R channel; second row left-right: AHE on B channel, AHE on G channel, HE on RGB channels; third row left-right: HE on RGB channels individually

By this cropping, the shape of the images is changed from a rectangle 640-by-480 to a square 368-by-336. With square images (width and height are almost equal, with a ratio of $368/336 = 1.09$), width and height of the images can be rescaled to the same number without distorting the images, as illustrated in Figure 17.

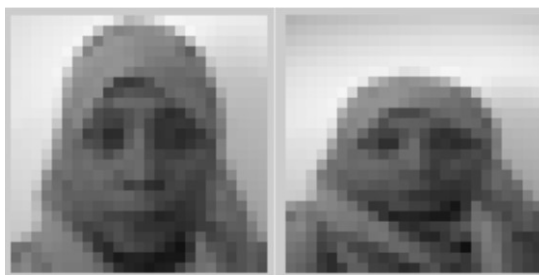


Figure 17: Image shape not distorted after rescaling to 25-by-25 due to cropping (left); image shape distorted after rescaling to 25-by-25 when cropping is not done since original images captured by webcam are not square (right)

The images are then rescaled to 25-by-25 resolution for simplicity. It will be easier to insert the values of all pixels into the neural network as input pixel when the pixel number is reduced from 123,648 (368-by-336) to only 625 (25-by-25). Thus, prior to training the neural network, each image undergoes the Figure 18 steps:

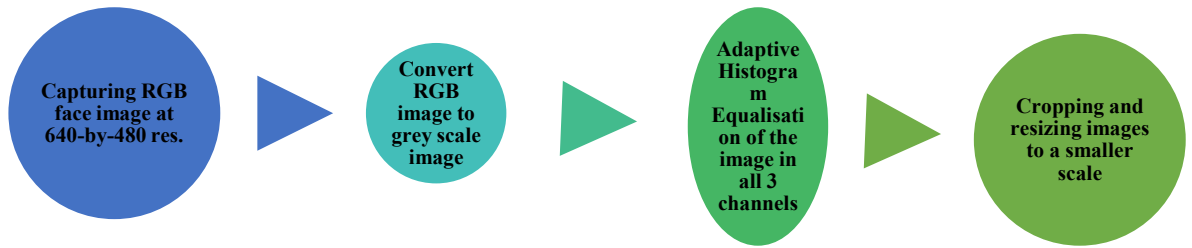


Figure 18: Steps of processing images prior to insertion into network as training input

Each processed image is inserted into the network as a training sample input. Insertion of images into the program is done by entering the pixel values of each grey scale image, one by one into one row. Since each image has a dimension of 25-by-25 image, there are a total of 625 pixel values in each grey scale image to be inserted, as shown in Figure 19, and 20:

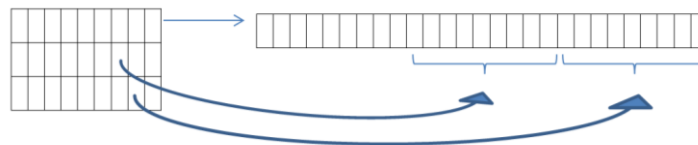


Figure 19: Displaying how image pixels inserted row-by-row as input into network

48	153	210	211	214	227	91	57	104	117	119	158	179	172	248	155	132	140	87	68	40	87	139	174	80
87	144	206	210	213	217	68	82	124	120	108	120	143	137	117	128	134	142	96	66	53	43	180	171	77
47	144	206	208	212	209	53	78	153	145	131	140	148	141	135	140	139	135	105	61	54	45	158	180	83
45	140	206	208	213	217	61	114	154	98	96	129	145	141	149	149	135	124	123	63	51	41	130	186	86
49	137	204	207	215	196	72	140	123	75	63	48	101	124	116	64	61	61	113	64	41	30	90	102	85
44	128	203	207	217	184	65	137	147	138	138	61	73	154	62	41	96	94	102	77	84	124	187	168	88
44	125	200	204	216	184	62	137	153	139	139	72	91	136	59	62	157	118	113	75	120	218	233	153	83
44	116	190	204	211	219	91	127	147	63	30	40	100	179	93	63	59	94	115	59	113	205	221	139	72
43	112	194	203	209	224	182	154	164	111	75	120	136	136	103	65	48	83	111	54	114	197	209	124	64
38	107	193	205	211	224	184	112	147	180	184	177	185	221	127	144	164	167	88	94	123	187	201	119	65
45	103	196	212	217	221	228	198	118	143	183	132	166	190	128	149	147	131	59	112	130	213	205	135	78
49	100	191	207	209	211	216	214	107	79	94	115	68	61	75	121	99	77	35	124	173	223	183	140	84
46	88	146	177	175	178	183	194	134	53	77	148	124	81	113	82	52	73	83	127	151	203	173	123	74
43	81	143	174	170	175	181	192	145	89	89	83	95	80	48	40	59	140	142	117	112	147	148	110	69
48	94	181	189	190	199	211	214	194	124	102	101	76	60	43	44	77	130	141	138	171	204	215	161	95
189	197	222	231	225	234	233	181	91	50	96	100	85	74	73	65	59	107	134	178	224	244	248	214	102
189	209	235	239	232	221	183	182	79	77	74	85	76	67	68	44	56	96	140	173	210	221	224	223	113
189	204	229	226	186	183	161	154	89	47	72	94	47	49	47	44	63	108	149	147	173	174	197	209	207
214	219	210	187	183	184	155	149	123	80	72	59	44	48	53	45	68	140	150	133	136	143	148	185	207
194	159	200	195	195	143	132	149	151	135	111	78	72	74	78	104	134	136	147	144	138	149	148	164	179
228	145	164	194	201	210	125	134	147	156	160	133	91	84	102	153	162	131	160	170	161	142	149	134	115
136	148	149	148	173	133	140	142	180	180	181	204	189	148	149	189	143	136	130	151	153	146	144	142	88

Figure 20: Total 625 (25 by 25) pixel values of processed grey scale image
As an example, pixel values are inserted row wise into one row, as such:

48	153	210	211	214	227	91	57	104	117	119	158	179	172
----	-----	-----	-----	-----	-----	----	----	-----	-----	-----	-----	-----	-----

After that, pixel values are normalised by division with 255 (highest possible pixel value) as follows:

0.1	0.60	0.82	0.82	0.83	0.89	0.35	0.22	0.40	0.45	0.46	0.61	0.70	0.67	0.57
-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------

A total of 34 face samples were used for training the network. The training samples consisted of faces of both male and female, adults (aged 20-50) and children (aged 5-8). The network is trained to detect faces by comparing the dark and bright spots in the faces. The areas protruding inwards around the eyes and down across the sides of the nose are usually dark, while the areas protruding outwards like the cheeks are brighter in any human face irrespective of the whether the eyes are closed or opened. Thus, the total input file for training the network is a 34×625 matrix consisting of 34 samples of 625 elements each. A sample training file is shown in Figure 21. Participants were made to close and open eyes for capturing two images of each participant. With this, the network will be trained to detect faces irrespective to whether eyes are opened or closed.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
1	0.2314	0.2980	0.5765	0.8627	0.6000	0.5176	0.2941	0.6392	0.8353	0.8196	0.4627	0.3059	0.5176	0.5294	0.4275	0.2667	0.4314	0	
2	0.2471	0.3725	0.6235	0.8471	1	0.8275	0.3255	0.3451	0.2353	0.2667	0.1804	0.2902	0.5059	0.5686	0.5490	0.5451	0.4471	0	
3	0.3137	0.4667	0.7373	0.9569	1	1	1	0.6588	0.4196	0.3098	0.3412	0.3647	0.2392	0.2588	0.3804	0.4000	0.2824	0	
4	0.3259	0.4431	0.6588	0.9137	1	1	1	0.9333	0.5333	0.4353	0.3647	0.3725	0.3020	0.2588	0.3333	0.3608	0.3608	0	
5	0.3329	0.6827	0.8784	0.8294	0.8510	0.3294	0.2745	0.2392	0.3020	0.1843	0.2588	0.4588	0.5686	0.5294	0.4549	0.3922	0.2186	0	
6	0.3725	0.7098	0.8627	0.8588	0.9137	0.9647	0.6314	0.3176	0.6902	0.3020	0.3373	0.5804	0.6824	0.6627	0.6314	0.6000	0.4510	0	
7	0.3961	0.7529	0.9725	1	1	0.6667	0.5098	0.6157	0.7451	0.4667	0.5922	0.6941	0.7647	0.7490	0.7294	0.7451	0.7098	0	
8	0.6980	0.7765	0.8510	0.9294	1	0.7412	0.3961	0.4392	0.4588	0.4588	0.4275	0.4039	0.3373	0.4471	0.6392	0.5490	0.3137	0	
9	1	1	1	0.9686	0.7216	0.4078	0.2980	0.4667	0.4196	0.4196	0.4941	0.6510	0.7216	0.7569	0.7882	0.7529	0.6314	0	
10	0.4039	0.4549	0.4980	0.5412	0.5529	0.5725	0.2392	0.0863	0.0824	0.0706	0.0667	0.0627	0.0471	0.1098	0.1725	0.1059	0.0431	0	
11	0.7451	0.8118	0.9373	1	1	1	0.6118	0.3176	0.2588	0.2706	0.3020	0.2902	0.2000	0.3020	0.4078	0.3216	0.1843	0	
12	0.9880	0.3529	0.6078	0.6000	0.5608	0.4941	0.0941	0.0824	0.0745	0.0745	0.0667	0.0510	0.0471	0.0471	0.0549	0.0627	0.0667	0	
13	0.3490	0.7369	1	1	1	0.9490	0.4627	0.3961	0.3686	0.3725	0.3490	0.2784	0.2353	0.1882	0.2353	0.2941	0.3176	0	
14	0.4431	0.4510	0.4745	0.5176	0.5098	0.1529	0.0667	0.0510	0.0549	0.0471	0.0392	0.0549	0.1216	0.1922	0.2118	0.2235	0.1647	0	
15	1	1	1	1	1	0.9765	0.5255	0.3333	0.2941	0.2745	0.2157	0.1725	0.2275	0.3922	0.5255	0.5451	0.5569	0.4392	0
16	1	1	1	1	1	0.9255	0.5490	0.4392	0.3882	0.3569	0.3137	0.3098	0.3333	0.3451	0.2941	0.2980	0.3666	0	
17	0.7373	0.4196	0.4784	0.6157	0.6471	0.3451	0.0706	0.0667	0.0588	0.0549	0.0471	0.0431	0.0510	0.0745	0.0824	0.0588	0.0471	0	
18	1	1	0.9961	1	1	0.9020	0.5098	0.3843	0.3412	0.3059	0.2941	0.2549	0.2784	0.3490	0.3922	0.3725	0.2745	0	
19	0.4196	0.5647	0.5569	0.5490	0.5922	0.6431	0.6667	0.6863	0.5961	0.1333	0.0627	0.0510	0.0863	0.2039	0.3098	0.2235	0.1412	0	
20	0.9765	1	1	1	1	1	1	0.9608	0.5137	0.3020	0.2000	0.2353	0.4353	0.5804	0.4314	0.2980	0.2980	0	

Figure 21: A section of the training input file

The network will be trained by supervision, known as supervised training. As such, the desired output will be provided to the network. The desired output values for each input image are four values that indicate the position of the nose with respect to the eyes, as illustrated in Figure 22.

These four values resemble the area surrounded by the eyes from the sides and surrounded by the eyebrows and nose tip from the upper and lower sides. These four output values are desired since the area in the middle of the two eyes can then be segmented and used for skin colour segmentation in the tracking phase.

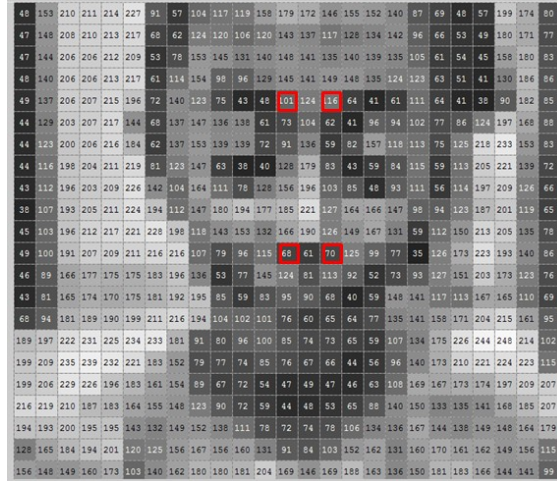


Figure 22: Input image displaying grey scale pixel values inserted into input file

Since the aim is to extract the position of the pixels and not the value, the output has been converted from grey scale value to the pixel position by the following formula:

$$\text{Output value} = [(\text{row number} - 1) \times 25] + \text{column number}; \quad (8)$$

For instance, in the above image, the four desired output values are those highlighted in red with grey scale values of 101, 116, 68 and 70. Since the input for each image has been inserted row wise in one row, the position of the four desired output values are retrieved by the following four equations:

$$\text{Position of pixel 101} = [(5 - 1) \times 25] + 13 = 113; \quad (9)$$

$$\text{Position of pixel 116} = [(5 - 1) \times 25] + 15 = 115; \quad (10)$$

$$\text{Position of pixel 68} = [(12 - 1) \times 25] + 13 = 288; \quad (11)$$

$$\text{Position of pixel 70} = [(12 - 1) \times 25] + 15 = 290; \quad (12)$$

Thus, the desired outputs are those pixels in the 113th, 115th, 288th and 290th column of the input 625-in-total pixels. Finally, the output values are normalised by division with 625 (highest output position possible is the last row number), to get the following:

0.1808000000000000	0.1840000000000000	0.4608000000000000	0.4640000000000000
--------------------	--------------------	--------------------	--------------------

Thus, the output file is 34×4 matrix consisting of 34 samples of 4 elements each, as shown in Figure 23:

	1	2	3	4	5
1	0.1808	0.1856	0.6208	0.6256	
2	0.1824	0.1856	0.5824	0.5856	
3	0.2224	0.2272	0.5424	0.5472	
4	0.2224	0.2272	0.5024	0.5072	
5	0.1808	0.1840	0.5808	0.5840	
6	0.1408	0.1456	0.5408	0.5456	
7	0.1808	0.1856	0.5808	0.5856	
8	0.2208	0.2256	0.5408	0.5456	
9	0.2224	0.2272	0.5824	0.5872	
10	0.2224	0.2256	0.5424	0.5456	

Figure 23: A section of the output file in training

After preparing the input and output files the network to be trained is designed. The neural network is a two layer (one Hidden layer and one Output layer) feed-forward network trained with the Levenberg-Marquardt and scaled conjugate gradient backpropagation algorithm. While 34 samples were inserted for training, 27 (80%) samples were used for training and 5 (15%) used for validation and 2 (5%) used for testing. The hidden layer consists of 20 neurons, set by default. The hidden layer works with a sigmoid function and the output layer implements a linear function, as shown in Figure 24.

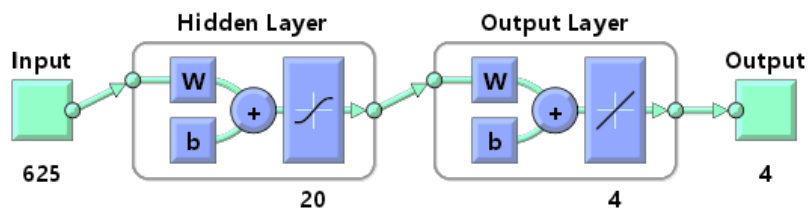


Figure 24: A rough diagram of the neural network to be trained

When simulating the network, the hidden and output layers will look like those shown in Figure 25:

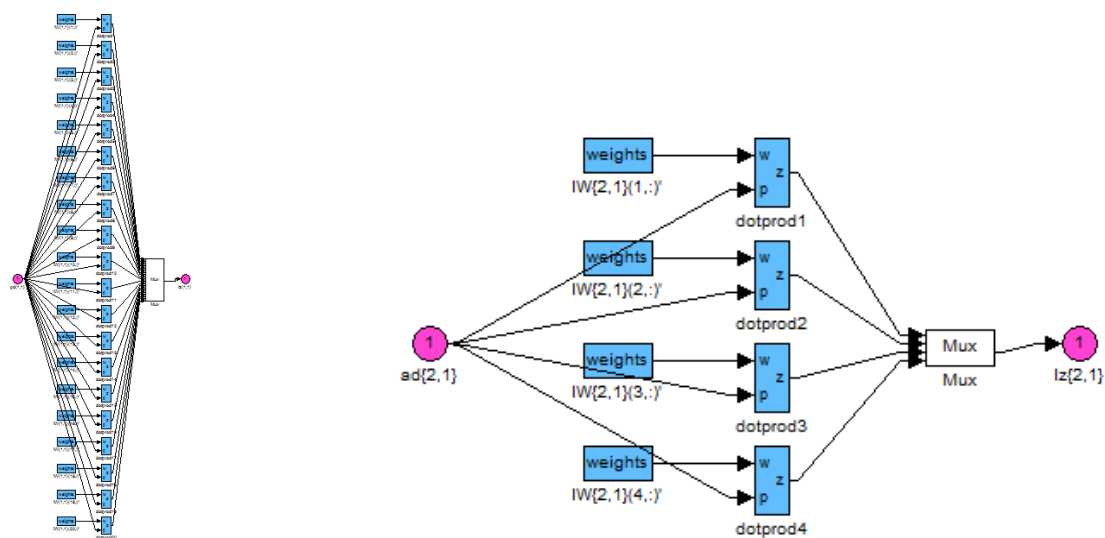


Figure 25: Network hidden layer, 20 neurons (left); network output layer, 4 neurons (right)

For training, 1,030 iterations were conducted. Each time the program stopped iterating since minimum mean squared error was reached, training was continued until 1,030 iterations were completed. This was done to ensure that the best value of error is achieved. The following Table 1 summarises the MSE (Mean Squared Error, the average squared difference between the outputs and the targets) and Regression (correlation between outputs and targets):

	Sample number	MSE	Regression
Training	27	0.00004632702398146551	0.994613

Validation	5	0.008716403854692	0.815132
Testing	2	0.001536056444238	0.999999

Table 1: Results of training data fitting network (20 hidden neurons)

The following graph in Table 2 illustrates the training results.

Graph	Comments
	Graph indicates how the error decreases with training, no overfitting occurred since test curve hasn't increased significantly above the validation curve
	Following graph shows the point at which training of the network stopped, and also shows how a minimum error was achieved after reaching 22 epochs of training.
	Regression of > 0.9 is noticed in all plots except for the validation plot, proving good match between the output and the desired target (a good fit is especially noticed in the training plot). The samples used for validation and testing were only 5 and 2, possibility of extrapolation (samples outside the training set), some of those are supposed to be included in the training samples, while another set of 5 and 2 samples needs to be used.

Table 2: Plotted results of training data fitting network (20 hidden neurons)

	Network Output				Desired Output				Error			
1	0.18	0.18	0.61	0.62	0.18	0.18	0.62	0.62	-	-0.004	.002	.0002
									.0001			
2	0.17	0.17	0.56	0.58	0.18	0.18	0.58	0.58	.003	.007	0.01	.001
3	0.22	0.22	0.53	0.55	0.22	0.22	0.54	0.54	-0.002	.003	.004	-0.004
4	0.22	0.23	0.50	0.51	0.22	0.22	0.50	0.50	-0.002	-0.003	-0.002	-0.004
5	0.18	0.18	0.57	0.58	0.18	0.18	0.58	0.5	-	-0.002	.0009	-0.003
									.0005			
6	0.18	0.35	0.51	0.62	0.14	0.14	0.54	0.54	-0.04	-0.20	0.02	-0.083
7	0.17	0.18	0.58	0.57	0.18	0.18	0.58	0.58	.003	-0.002	-0.002	.009
8	0.22	0.21	0.53	0.54	0.22	0.22	0.54	0.54	-	.010	.006	-
									.0004			.0004
9	0.30	0.48	0.62	0.75	0.22	0.22	0.58	0.58	-0.07	-0.257	-0.04	-0.16
10	0.22	0.21	0.53	0.53	0.22	0.22	0.54	0.54	.0001	0.01	.008	.005

Table 3: Measured vs. Desired output values of trained data fitting network (20 hidden neurons)

Thus, the network has been trained to detect faces in front of the camera based on a certain sequence of dark and bright spots that is same in every face.

Training Data Fitting Neural Network (with 125 Hidden Neurons)

Input and output files are same as the previous network (input file is a 34×625 matrix consisting of the image pixel values inserted row-wise and output file is a 34×4 matrix consisting of the four required positions of the nose area). 1000 epochs were completed in training. Following are the resulting graphs:

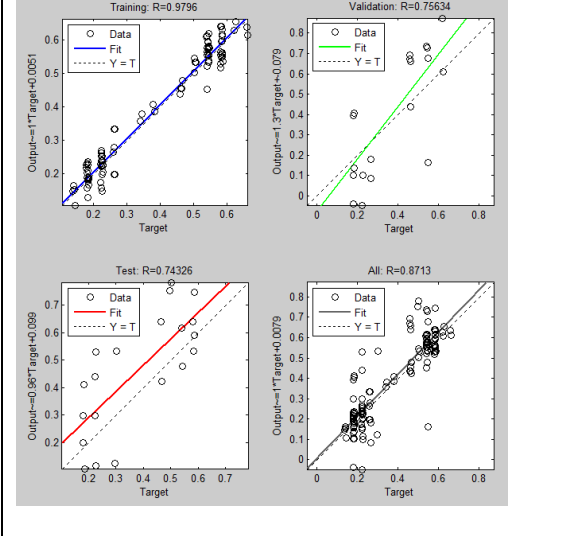
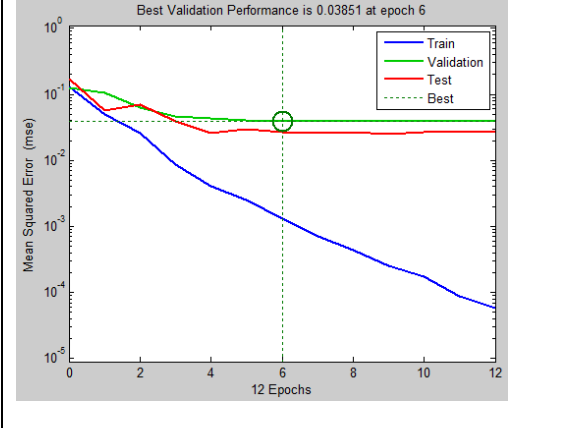
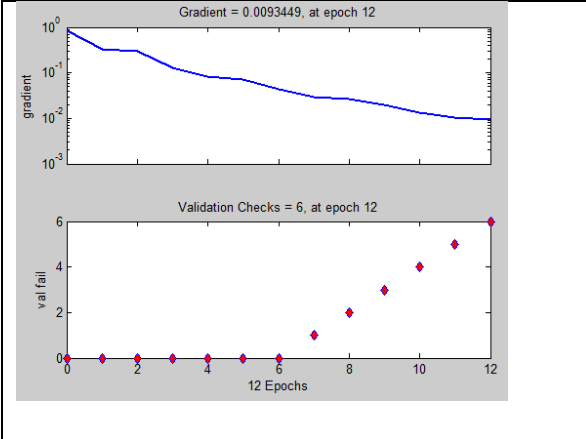
Graph	Comments
	<p>Regression value for the training set is satisfactory, but the validation and test sets show quite low values for the regression slope</p>
	<p>MSE (Mean Square Error) is not very low, overfitting about to happen</p>
	<p>Following training state values are plotted until the point at which training stopped.</p>

Table 4: Plotted results of training data fitting network (125 hidden neurons)

The following Table 5 summarises the training results:

	Sample number	MSE	Regression
Training	24	0.0013327	0.853354
Validation	5	0.03851	-0.272043
Testing	5	0.0266861	-0.406954

Table 5: Results of training data fitting network (125 hidden neurons)

Training Pattern Recognition Neural Network (with 20 Hidden Neurons)

The input file is a 39×625 matrix consisting of the image pixel values. 34 samples are faces and 5 samples are non-faces (random photos). Values of the input file were inserted row-wise. The output file is a 39×1 matrix, each entry consisting of 1 or 0. 1 resembles a face and 0 resembles a non-face. 1000 epochs were completed in training.

The following Table 6 summarises the training results:

	Sample number	MSE	% Error
Training	27	0.0000000388487	0
Validation	6	0.0000554176	0
Testing	6	0.000116322	0

Table 6: Results of training pattern recognition network (20 hidden neurons)

The following are the resulting graphs, shown in Table 7.

Graph	Comments
	<p>27 samples were used for training, 6 samples were used for validation, 6 samples were used for testing. No misclassifications are noticed.</p>
	<p>Overfitting observed since test curve rises beyond validation curve.</p>
	<p>Following training state values are plotted until the point at which training stopped</p>
	<p>A 90 degree angle notifies good recognition results</p>

Table 7: Plotted results of training pattern recognition network (20 hidden neurons)

Training Data Fitting Neural Network (with 20 Hidden Neurons)

The difference between this network and the first one trained with 20 hidden units also, is that here image pixel values of the input file were inserted column-wise (as illustrated in the Figure 26). This method was tried to check if the network can recognise faces better, since the previous network displayed correct row number but incorrect column number. The other difference is that the output file is 34-by-4 matrix, consisting of 34 samples of 4 desired face points that has not been normalised.

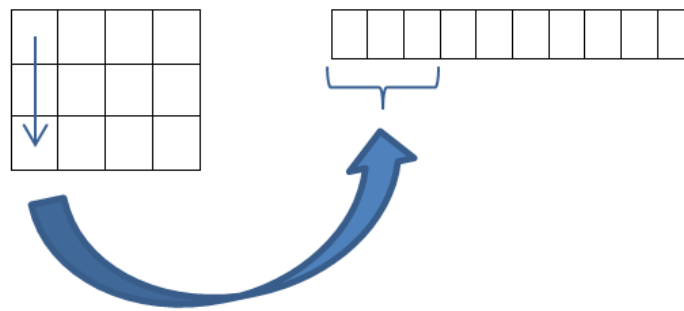


Figure 26: Inserting pixel values column-wise

Once, 1000 epochs were completed in training, the following are the resulting graphs, as shown in Table 8:

Graph	Comments
	<p>Regression values are satisfactory in case of the training samples, but unsatisfactory for the validation and testing samples.</p>
	<p>MSE (Mean Square Error) of the validation and test samples is quite high, which is not desirable.</p>
	<p>Following training state values are plotted until the point where training stopped.</p>

Table 8: Plotted results of training data fitting network (20 hidden neurons), column-wise insertion

2.1.1.5 Method of Support Vector Machines

The concept of kernels was implemented in training support vector machines for face detection in the calibration stage. Different SVMs were trained, such as linear and kernels (Gaussian, polynomial), to compare the performance and choose the one with best results.

SVM with Linear Kernel

The input file is a matrix of size 39×625 , consisting of pixel values of 39 samples of both positive (face) and negative (non-face) samples. The groups file (like output file) is a column vector of size 39×1 , consisting of either 1 (face) or 0 (non-face). The following are the results after training, shown in Table 9:

Support Vectors	7 support vectors of indices: 9, 11, 14, 15, 16, 17, 19
Alpha	-0.0109238356105662,0.00602186490777510,0.00185224308926318,- 0.00167132347468738,-0.000650312134400455,- 0.00439243922084035,0.00976380244345609
Bias	-1.3530
Correct Rate	0.8157894736842105
Error Rate	0.18421052631578946

Table 9: Results of trained SVM (linear, 7 SV)

Another SVM was trained with different number of support vectors, but produced similar results, as shown in Table 10:

Support Vectors	14 support vectors of indices: 5, 11, 14, 16, 18, 20, 22, 24, 28, 30, 34, 36, 39
Alpha	0.0123170776675077,0.00798968508745530,-0.00190989789279458,- 0.00518185056588518,0.00298002723509351,-0.00755956823622462,- 0.00526700764447905,-0.00257968666905058,-0.00116720210229293,- 0.00467626061145311,-0.00496412222423166,-0.00233300095995789, 0.0129179054334112,-0.000566098517098042
Bias	-1.9389

Table 10: Results of trained SVM (linear, 14 SV)

Hence, training a linear SVM by the above does not separate the face from non-face images to the desired extent as seen from the error rate.

SVM with Gaussian (RBF) Kernel

Training a SVM with a Radial Basis Function (RBF) or Gaussian Kernel gives a correct rate of 0.8947, a value close to the previous SVM with a linear kernel.

SVM with Polynomial Kernel

Since previous trainings didn't produce the desired minimum error rate and maximum correct rate, detailed training needs to be done where the individual parameters are set manually. As such, the input file is a combination of two files. One part of the input file is a 34×625 matrix consisting of pixel values of face images (positive samples) inserted column-wise. The other part of the input file is a 34×625 matrix consisting of pixel values of random non-face images (negative samples) taken from the Internet. The grouping file consists of only 1's and -1's. 1's resemble positive samples and -1's resemble negative samples. 17 of the positive and negative samples were kept for training (a total of 34 training samples, half of which

are 1's and the other half are -1's), while the remaining 17 of each of the positive and negative samples were kept aside for validation of the training results.

Different number of support vectors was selected by the machine as well as different accuracy levels for both the training and validation stages depending on the degree of the polynomial of the kernel. This is shown in Table 11.

Polynomial degree for Kernel	Order One	Order Two	Order Three	Order Four
No. of Support Vectors	60	30	25	23
Training Accuracy	0.5	1	1	1
Validation Accuracy	0.5	0.75	0.875	0.875
Time Taken for training	0.0 sec	0.0 sec	0.0 sec	0.0 sec

Table 11: Comparing different degree polynomial SVM

As the degree of polynomial is higher, fewer number of support vectors are required. Also, the performance increases as the degree of polynomial is higher, as illustrated from the accuracy levels. However, performance reaches a maximum at polynomial degree three after which it doesn't increase further. Thus, a kernel polynomial of degree three was selected for face detection in the calibration phase.

After training, values of alpha were calculated by the kernel classifier, after which the weights can be calculated by the equation:

$$w = \sum_i \alpha_i y_i x_i$$

The weight matrix is of size 1-by-625. From this weight matrix, and the bias calculated by the kernel classifier during training, the confidence level of every point can be calculated by the equation:

$$y_i(\langle w, x_i \rangle + b) \geq 1$$

To check for the working of the classifier, the weight matrix is plotted, which looks like an average face from the samples, shown in Figure 27:



Figure 27: Plotted SVM weight matrix

This plotted weight matrix shows that the kernel classifier was successfully trained to detect faces and non-faces. The value of the constraint C affects the face detection results. As such, different values of C were tested and results are shown in Table 12:

Value of C	1000	100	10	1	0.1	0.01	0.001	0.0001	0.00001	10^{-8}
No. of SVs	10	10	10	12	12	12	12	12	15	33
Training Acc.	1	1	1	1	1	1	1	1	1	1
Validation Acc.	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.647	0.676	0.68

Table 12: Comparing SVM performance with different C values

When plotting the weight matrix after training with different values of C , it can be noticed that lower the value of C , more the weight matrix looks similar to a face, as shown in Figure 28:

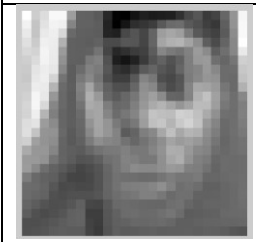

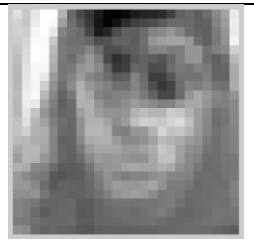
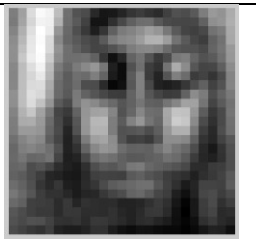
Weight Matrix with C=1000:	Weight Matrix with C=1:	Weight Matrix with C=0.001:	Weight Matrix with C=0.00000001:
			

Figure 28: Plotted weight matrix for different C values

Thus, $C=0.00000001$ was selected as the final value for training a face detection SVM in the calibration phase.

A couple of support vector machines were trained with different sets of face positive samples. The final SVM selected with best results produced is a 20 positive face samples and a 20 non-face samples. The samples have been properly selected to represent different faces (male and female, adults and kids). The training samples were cropped to a 50-by-62 size without distorting the height/weight ratio of face images. Thus, the weight matrix is of size 1-by-3100. The results are shown in Table 13.









	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	Gaussian kernel
Support vectors no.	20	20	20	20
Training samples accuracy	0.5	0.5	1	0.5
Validation samples accuracy	0.5	0.5	0.55	0.5
Weight matrix plotted				
Av. Weight matrix plotted				

Table 13: Comparing SVM performance for different kernels

2.1.2 Feature Extraction

After eyes are detected by using the first three methods discussed above, the final result is the black and white image, as shown in Figure 29. From this image, multiple features can be extracted that will help in the tracking phase.



Figure 29: Eye detected in b/w image

Extracting eyes position:

After the black and white image is extracted, the size of the eyes is extracted similar to Figure 30. Extraction of these measurements is helpful later on for tracking the eyes by feature matching.

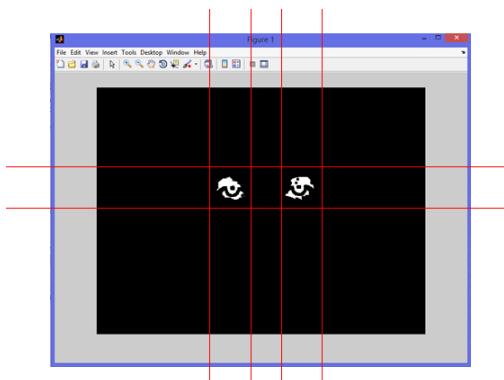


Figure 30: Extracting eye measurements

For each of the right and left eye, a search is done for the x and y coordinates of the eye (basically a search for the x and y coordinates of the white pixels):

$$\text{Height of eye} = \text{Maximum } y \text{ coordinates} - \text{Minimum } y \text{ coordinates} \quad (13)$$

$$\text{Width of eye} = \text{Maximum } x \text{ coordinates} - \text{Minimum } x \text{ coordinates} \quad (14)$$

These equations give an approximation of the number of pixels occupied by each eye in the captured images.

Extracting eyes colour:

The values of the three colour channels of the RGB colour space for the eyes are calculated. This is done by extracting the mean and standard deviation of the three colour channels in the eye area detected. This will be helpful since different people have different colour of the iris. As such, deviations in the RGB channels are noticed to be different. This eye colour extraction can be used later on to confirm that

the location of the eye detected is correct by checking on the RGB values of that detected area. As inferred from Table 14, the different colours of the iris are reflected on the different RGB values of the eye area. However, other factors, like illumination, also affect the RGB values in the captured picture.





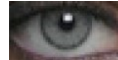
		Black	Brown	Green	Blue	Grey
						
Red Channel	Mean	30.3833	145.9635	110.2702	131.6454	88.3576
	Standard Deviation	20.6293	60.8245	66.8751	62.0764	41.4539
Green Channel	Mean	26.2451	107.7202	92.5414	128.1594	81.1568
	Standard Deviation	19.9083	59.7821	62.4807	57.4799	39.1077
Blue Channel	Mean	22.2741	90.8608	85.4102	144.9685	78.9416
	Standard Deviation	20.1335	65.2992	62.7050	58.6403	39.9367

Table 14: Different RGB values for eyes with different iris colour

In tracking techniques, features that are unique in all people are usually exploited for tracking purposes. Such is the case of the famous detection algorithms like Viola-Jones, where the dark spots that are common in all people's faces are exploited (known as the Haar-like features). As such, here, two features are exploited: the pupil and the cornea. Whatever be the colour of the eyes (the iris),

there will always be black and white pixels inside the eyes (which are the pupil and cornea respectively). These two features, illustrated in Figure 31, are common in all people.

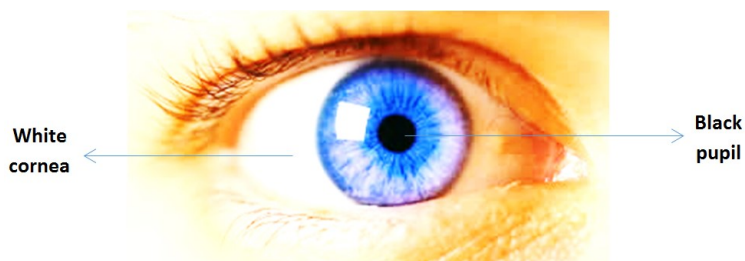


Figure 31: Human eye (Shutterstock.com, 2015)

In RGB colour space, black has a value of 0 and white is 255. Other colours are between 0 and 255. In tracking the white and black pixels, first attempt was to set a fixed threshold to detect the black and white pixels. As such, the system was designed to consider any pixel above 130 as white. The threshold value was chosen based on trial-and-error method on a sample picture. However, the program couldn't detect the white cornea in many other samples. Thus, it was found that setting a fixed threshold for the black and white was not practical since the work is in RGB colour space. RGB values change with illumination, due to which the white cornea pixels might have values below 130. And other attempts to increase or decrease the thresholds didn't work for all testing attempts. As such, another method needs to be implemented that is universal and not constrained by a threshold value. Thus, histogram equalization was chosen.

In histogram equalization, the contrast is enhanced and induced boundaries are eliminated. Thus, by this technique, white pixels can be easily distinguished from black pixels. However, this contrast enhancement is applied to the red channel only of the eyes. It has been noticed that applying contrast enhancement on the red

channel produces best results with regards to distinguishing between white and black pixels. The difference is illustrated in Table 15. This has been explained by Majumder et al. (2011) from the paper ‘Directional and nondirectional spectral reflection from the human fovea’ by Kraats and Norren (2008). They explained that there exist red blood vessels behind the retina. Also, red colour exhibits the highest wavelength compared to other colours. Thus, they worked on the red colour space represented in hue. The figures in Table 15 prove that red channel produces best results of contrast between white cornea and black pupil in the RGB colour space.

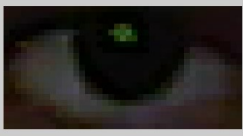





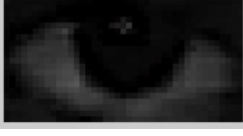

	Before Histogram Equalisation	After Histogram Equalisation
RGB Channel		
Red Channel		
Green Channel		
Blue Channel		

Table 15: Histogram Equalisation on the different colour channels

After histogram equalization, a search is done for the maximum (highest value) and minimum (lowest value) pixel in the eye area. This is done under the assumption that the darkest pixel will correspond to the black pupil, while the brightest pixel will correspond to the white cornea, as illustrated in Figure 32.

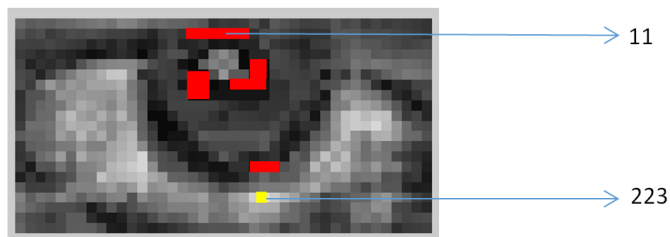


Figure 32: Brightest pixels in yellow (cornea), darkest pixels in red (pupil)

In the above figure, the maximum (highest pixel value) and minimum (lowest pixel value) are shown in yellow and red along with their pixel values. Also, the standard deviation of the red channel of the eye area after histogram equalization is extracted. The extraction of this standard deviation is due to the fact that the cornea region exhibits different pixel values that are close to white, and the pupil region exhibits different pixel values that are close to black. Uneven surrounding lighting is the main reason for these different pixel values, as well as other reasons like the camera quality. As such, knowing the deviation with respect to the mean will help the system detect most of the pixels in the cornea region and that in the pupil region.

This eye contrast adjustment (by histogram equalization) as well as measurement of the white pixels and the black pixels, will be used later for tracking eyes.

Extracting skin colour:

Skin colour segmentation is the major step for face detection in this research work. As such, values of the RGB colour channels of the skin can be determined after the eyes are detected in the calibration phase. The area between the two eyes is used for skin colour extraction, as shown below in Figure 33.

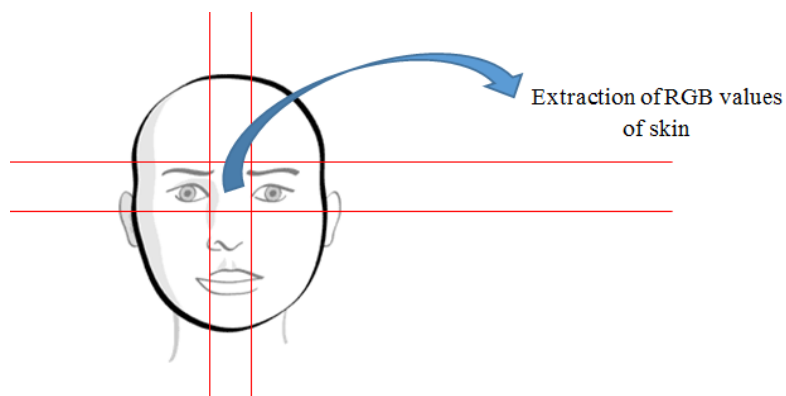


Figure 33: Extraction of RGB values of selected skin area

Extracting facial measurements:

The steps in this program run in a sequential way. The RGB values of the skin, extracted previously, are exploited for detection of the entire face by a simple search of the RGB values in the image captured. The pixels with RGB values in the range of $(\mu + \sigma)$ and $(\mu - \sigma)$ are detected as the face. Other pixels misclassified as face are removed by morphological operations. Thus, the skin colour extracted previously from a small area between the eyes is used for whole face detection. The method worked successfully for different faces, including people with beard. The result is shown in Figure 34.

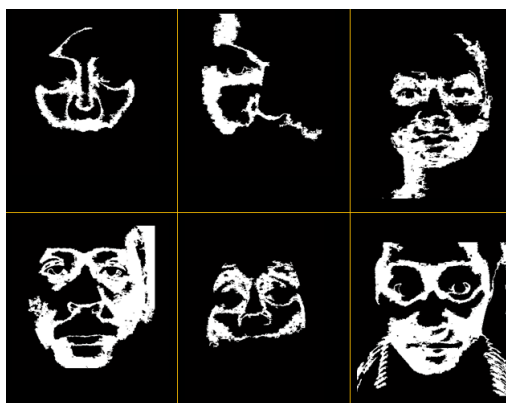


Figure 34: Face detection by skin colour segmentation

From this detected face, the facial measurements of the forehead, chin, left side and right side are calculated, as shown in Figure 35. These measurements will be helpful later on when tracking the eyes in the face.

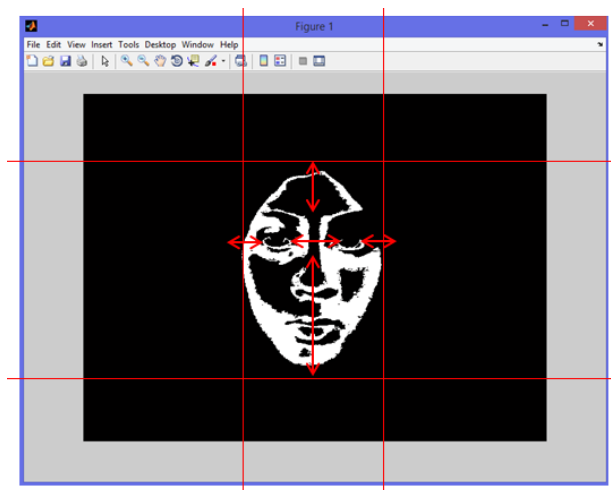


Figure 35: Extracting face measurements

2.2 Eye Tracking

Using the features extracted, tracking has been implemented in a couple of ways involving different combinations of features extracted and that combination with best results have been chosen finally. In all combinations, skin colour segmentation has been used as a main technique for tracking the user's face.

2.2.1 Combination 1: White and Black Pixels in Eyes

Here, after the calibration stage and feature extraction, the eyes are tracked by feature matching in the following steps, as shown in Figure 36.

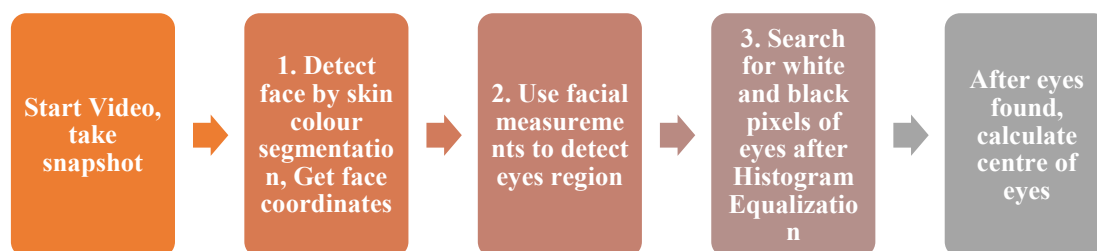


Figure 36: Steps of tracking combination 1

Steps 1, 2 and 3 consist of feature matching with those extracted from the calibration stage. For step 3, the minimum and maximum extracted before in the calibration phase are utilised for tracking the pupil and cornea. The maximum (highest pixel value) is assumed to be the white cornea while the minimum (lowest pixel value) is assumed to be the black pupil. The eye area, shown below in Figure 37, is transformed to its red channel and histogram equalization is performed. Next step is to search for the cornea and pupil, using the conditions:

Pupil detected if:

$$\text{pixel value in the range of } (\text{minimum} \pm \text{st.dev. of equalised red channel of the eye}) \quad (15)$$

The following Figure 37 will be the result. As can be noticed, only the pupil and the eye lashes are detected.



Figure 37: Dark pixels detected

Cornea detected if:

$$\text{pixel value in the range of } (\text{maximum} \pm \text{st.dev. of equalised red channel of the eye}) \quad (16)$$

As a result, only cornea is detected, while other single pixels are removed by simple morphological operations, as shown in Figure 38.



Figure 38: Bright pixels detected

The system works by detecting both the pupil and the cornea ((15) & (16)), as shown below in Figure 39. This is because it was found that relying only on dark pixel detection or only on bright pixel detection might lead to false detection of the eyes. However, relying on both detections increases the chance of correct eye detection. After tracking the pupil and cornea, centre of detected (white + black pixels) is regarded as centre of the eye. The following Figure 39 shows the tracked centre of eye.



Figure 39: Calculating centre of both dark and bright pixels

If results match (black and white pixels that fall in the specified range of (15) and (16) are found), eyes are detected and marked (small coloured rectangular) as shown in Figure 40.

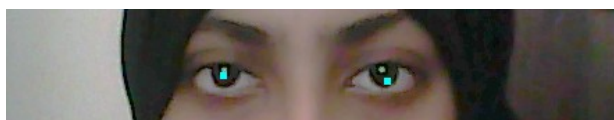


Figure 40: Detected Eye Centres

2.2.2 Combination 2: Mathematical Face Measurements

This tracking method has been implemented after face is detected with the trained support vector machine. Skin colour of detected face is extracted. This skin colour is utilised to detect the face while tracking. Skin colour segmentation results in a black-white image. Here, histogram equalisation on the red channel is done, after which the search for the black pupil (minimum RGB values point) is implemented. The sequence of stages is shown in Figure 41.

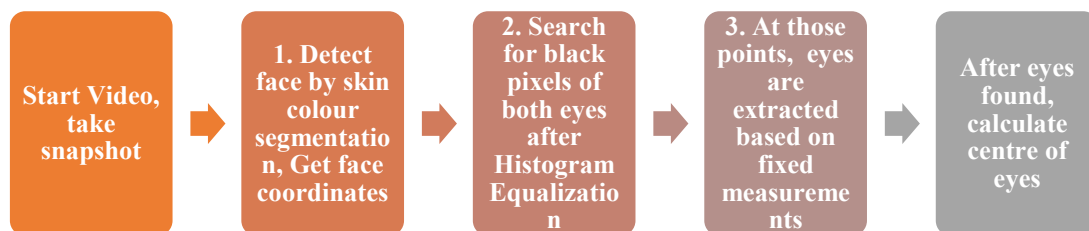


Figure 41: Steps of tracking combination 2

In the above step 3, fixed measurements are calculated based on known facts in every person:

$$\text{Eye width} = \frac{1}{4th} \text{ of face width} \quad (17)$$

$$\text{Eye height} = 0.6 \times \text{Eye width} \quad (18)$$

2.2.3 Combination 3: Eye Detection SVM

Another technique chosen and implemented is the one stated in Figure 42. It came up with best results as illustrated from the testing results in chapter 3. This is because this tracking method has been implemented after studying the drawbacks and false results from previous tracking methods. The face is detected with the trained support vector machine, after which skin is extracted.

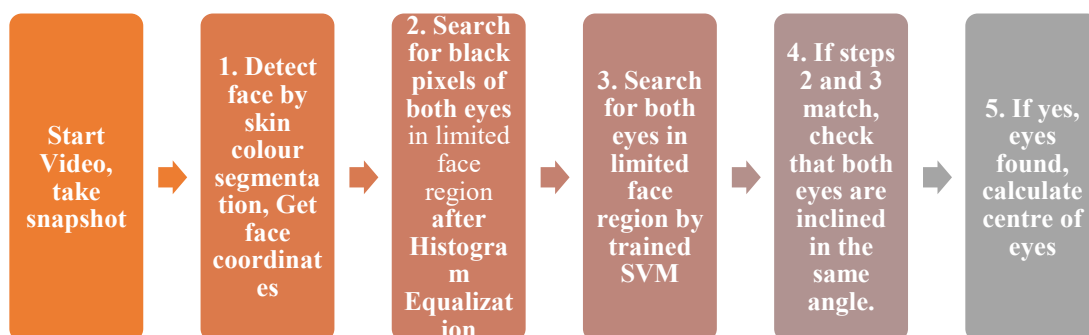


Figure 42: Steps of tracking combination 3

This combination of methods is unique due to step 4 and 5, in which two checks are implemented to verify if eyes are detected correctly or not. As an illustration, the steps are tabulated in Table 16 as follows:



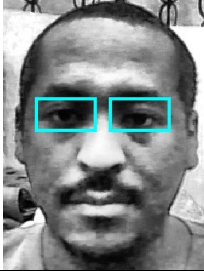
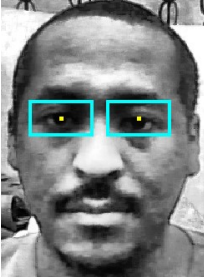

Start Video, take snapshot	
1. Detect face by skin colour segmentation, get face coordinates	
2. Search for black pixels of both eyes in limited face region after Histogram Equalisation	
3. Search for both eyes in limited face region by trained SVM	
4. If steps 2 and 3 match, check that both eyes are inclined in the same angle.	
5. If yes, eyes found, calculate centre of eyes	
Eyes detected!	

Table 16: Tracking steps in combination 3 (visualised)

In step 3, a number of support vector machines were trained to detect eyes. Training was done in the same way as face training during the calibration phase, where both eye and non-eye samples are fed into the support vector machine. The eye samples are represented by 1's and the non-eye sample images are represented by -1's. The support vector machine, with the help of support vectors, forms a margin by equation

(2) to distinguish between eye and non-eye samples by satisfying equation (3).

Different SVMs were trained, and are listed in Tables 17-20.

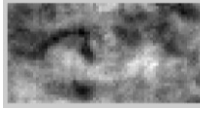
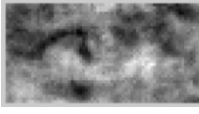

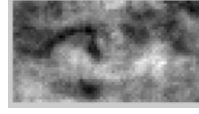
20 eye 20 non-eye samples, all 35-by-70	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	Gaussian kernel
No. of support vectors	20	20	17	20
Training samples accuracy	0.5	0.55	1	0.5
Validation samples accuracy	0.5	0.5	0.5	0.5
Weight matrix plotted				

Table 17: SVM Eye training results (35-by-70 dimension, 40 training samples)

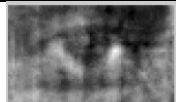



24 eye 24 non-eye samples, all 45-by-75	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	Gaussian kernel
No. of support vectors	19	10	11	24
Training samples accuracy	0.833	1	1	0.5
Validation samples accuracy	0.875	0.792	0.75	0.5
Weight matrix plotted				

Table 18: SVM Eye training results (45-by-75 dimension, 48 training samples)






26 eye 26 non-eye samples, all 35-by-65	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	5 th degree Polynomial kernel	Gaussian kernel
No. of support vectors	25	25	20	11	26
Training samples accuracy	0.5	0.5	0.654	1	0.5
Validation samples accuracy	0.5	0.5	0.577	0.654	0.5
Weight matrix plotted					

Table 19: SVM Eye training results (35-by-65 dimension, 52 training samples)

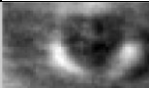

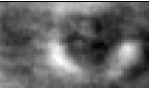
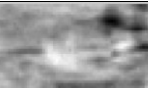
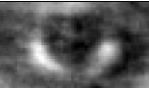
50 eye 50 non-eye samples, all 35-by-65	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	4 th degree Polynomial kernel	Gaussian kernel
No. of support vectors	49	49	38	22	50
Training samples accuracy	0.5	0.5	0.74	1	0.5
Validation samples accuracy	0.5	0.5	0.3	0.48	0.5
Weight matrix plotted					

Table 20: SVM Eye training results (35-by-65 dimension, 100 training samples)

Different parameters for the dimensions of the image samples were selected, such as 35-by-65, 45-by-75 and 35-by-70. This was done to check which dimension will work the best. The dimension decides how much area of the eye of the person will be cropped to be used as training sample. The dimension is also equal to the size of the window that is used after training to detect eyes of the user. The user's face image is divided into blocks. This window works by moving from one block to another and detecting if the eyes are present or not. If eyes in the captured images are bigger than the window size, chances are that eyes will not be detected (since the eyes fall out of the block). Hence, the aim is to set a small size for the window since calculations will be easier, but also an appropriate size that can fit user's eyes into a single block. Two-eye detection SVM was also trained to test for best performance, as follows in Table 21:





12 eye 12 non-eye samples, all 35-by-175	1 st degree Polynomial kernel	2 nd degree Polynomial kernel	3 rd degree Polynomial kernel	Gaussian kernel
No. of support vectors	9	11	9	12
Training samples accuracy	1	1	1	0.5
Validation samples accuracy	0.667	0.75	0.583	0.5
Weight matrix plotted				

Table 21: SVM Two-eye training results

The search in step 3 for the eyes is performed in a way similar to a moving fixed-sized window. The image is divided into sub windows. In each window, the confidence level is calculated using the weight matrix of trained SVM. If confidence level exceeds a set threshold, eyes are assumed to be detected and the position of that window is extracted. This window SVM method has been implemented by INRIA Visual Recognition and Machine Learning in Summer School (July, 2012). The limited face region where search for both eyes is performed is in the face centre where the eyes are located as per the human face golden ratio (Holland, 2008), pictorially shown in Figure 43.



Figure 43: Eye search area (face golden ratio)

2.3 Classification

To make the keyboard as simple as possible, only three movements of the eye ball are detected, right, left and centre respectively. The method of dark pixel detection discussed under the tracking combination algorithm 2 is implemented. As such, the eye appears as in Figure 44. Calculation of the position of the eye ball is done by calculation of the ratio of white to black pixels in different positions of the tracked eye to locate the eye pupil. Right or left positions of the eyeballs indicate that a key is pressed. Centre indicates that no action is to be taken. This is further illustrated in Figure 45.

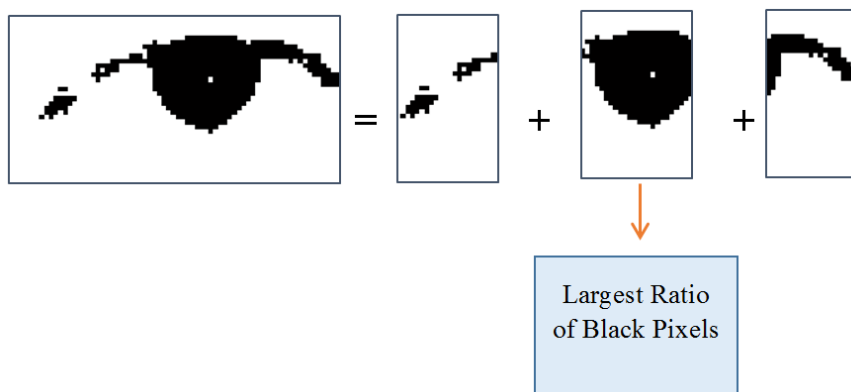


Figure 44: Detection of eye ball position during classification stage

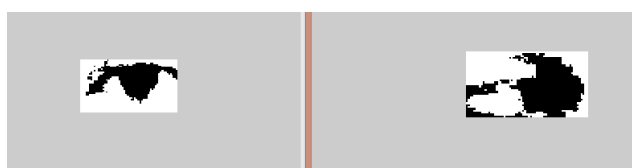


Figure 45: Eye indicating pressing key (right); eye indicating no key to press (left)

The eye tracking keyboard can be designed to detect a variety of different eye movements. The aim, here, is to make the keyboard as simple as possible. The keyboard is based on the technique of Partner-Assisted Scanning, as discussed in chapter 1. PAS is a known communication technique for those with difficulty in speaking/moving. PAS keyboards can be designed in a varied number of ways. They could display letters, sentences or alphabet. They can be designed in different designs and languages. As such, the keyboard scans through each key as shown below in Figure 46.

1	2	3	4	5	6	7	8
A	B	C	D	E	F	G	SPACE
H	I	J	K	L	M	N	
O	P	Q	R	S	T	U	
V	W	X	Y	Z	9	0	

Figure 46: Designed PAS keyboard

The yellow square moves through each letter, pauses for one second and then moves to the next letter and so on. The speed of the pause can be customised to match the pace of the user. This set up is suitable for people with paralysis and physical disabilities. When the desired letter is highlighted, the user gestures by moving her/his eyes to one side. When the system detects that the user's eye ball has been moved sideways, that highlighted letter is printed on the screen.

The tracking program and scanning keyboard work independently of each other. This is due to the single thread nature of the program used. Overall, three files run simultaneously. One is the tracking system that captures images continuously and searches for the eyes in each image. Second program that runs simultaneously is the keyboard that scans through each letter continuously by highlighting the letters on by one in a fixed time interval between each letter. This program is independent and only displays the scanning keyboard. It does not interfere with the working of the eye tracking system. Third is another program (a function), that sends the key to be displayed to the main program. Since the keyboard scans letters independently, the selection of the key to be displayed is done with the help of a timer. The keyboard scans at a fixed time of one second per letter. There are 40 letters, so the entire keyboard is scanned in 40 seconds. Thus, a timer that calculates each round of image capture is used to select the key to display by the following simple equation:

$$key = remainder\ of\ \frac{time\ elapsed\ in\ seconds}{40} \quad (19)$$

Thus, the overall three files communicate as shown in Figure 47.

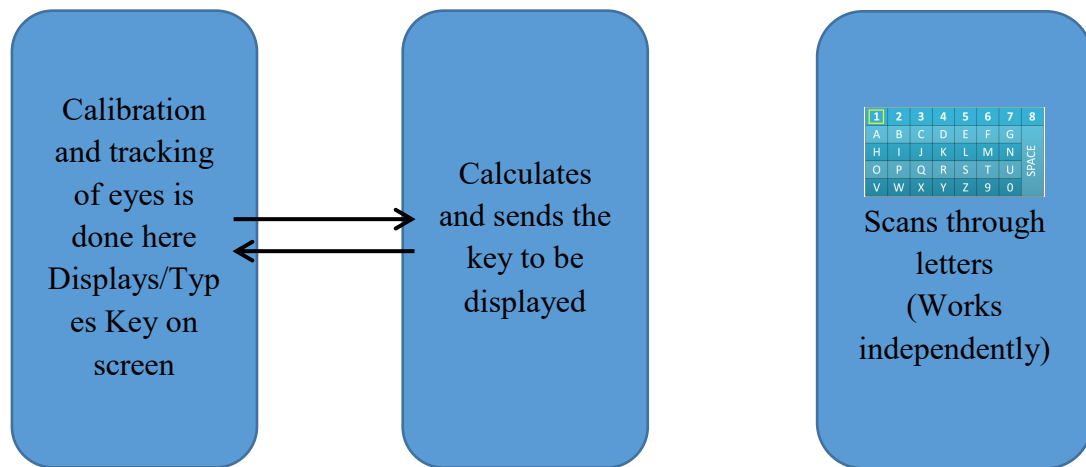


Figure 47: Communication between the three eye tracking system files

Thus, the system works as shown in Figure 48:

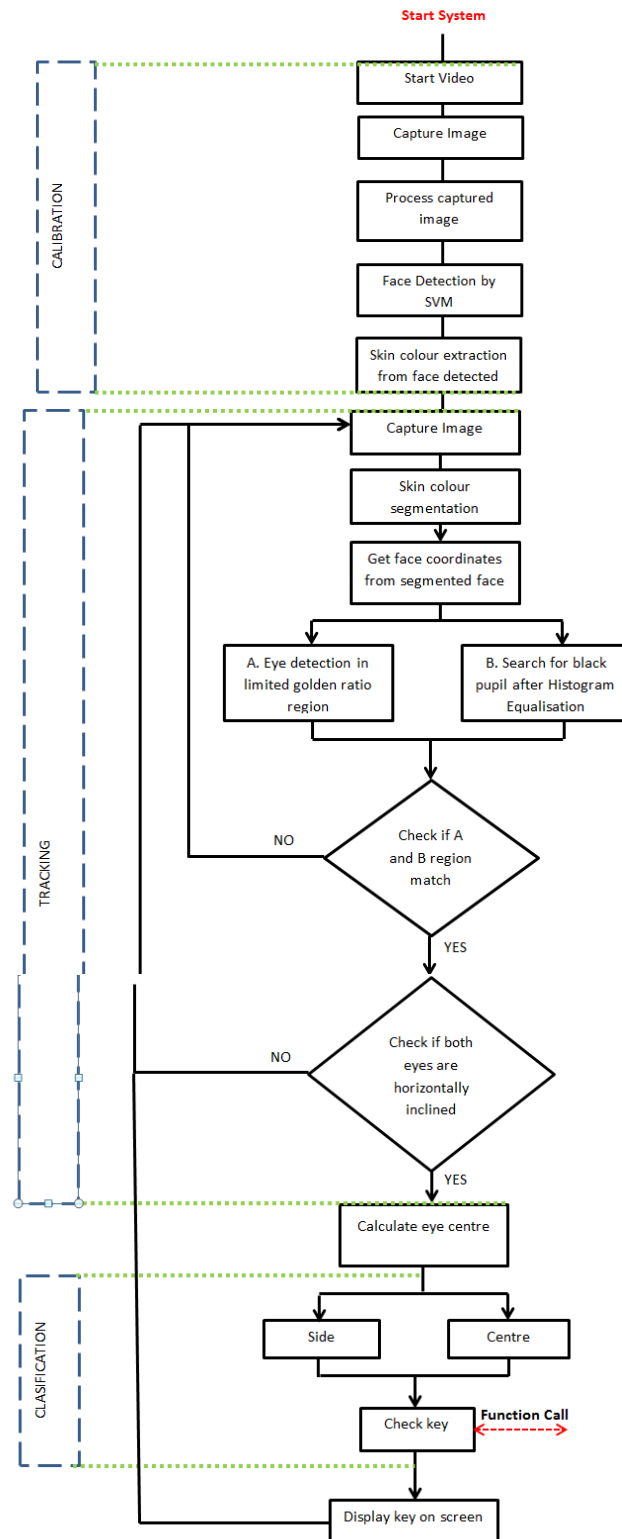


Figure 48: Flow chart of PAS eye tracking system

Chapter 3: Results

The results were obtained by testing the built programs. Both unit testing and system testing were performed.

3.1 Testing of Calibration Phase

Multiple Frames

Results of this method are quite satisfactory. However, the existence of multiple frames is cumbersome. Testing was done multiple times on the same individual. On testing, results showed that sometimes system fails to select the best image. The existence of maximum number of white pixels in some images might not indicate the position of the eyes in the image, as is the case in frame 7 of Figure 49. On the other hand, when the best frame is correctly selected, the results are accurate and exact position of the eyes in the image is located.

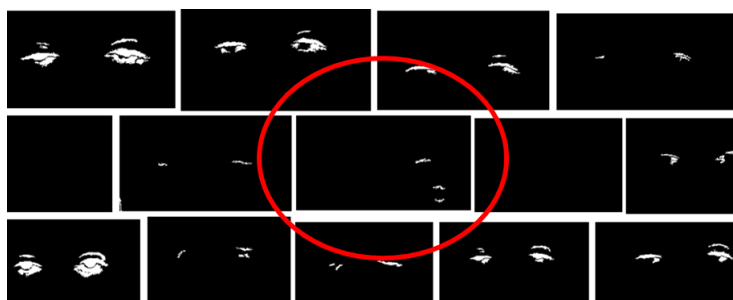


Figure 49: Frame 7 detects wrong pixels as eyes in multiple frames method

Bounding Box

Testing was done on 10 different individuals, both male and female, adults and kids. Sample results shown in Figure 50 proved efficiency of the method and ease of implementation. In one case, the eyes were not perfectly detected due to uneven lighting inside the room. Overall, testing results were successful by 96 % (24

successful attempts out of 25). One attempt failed to detect the eyes because of poor quality images taken by camera under very dim light.

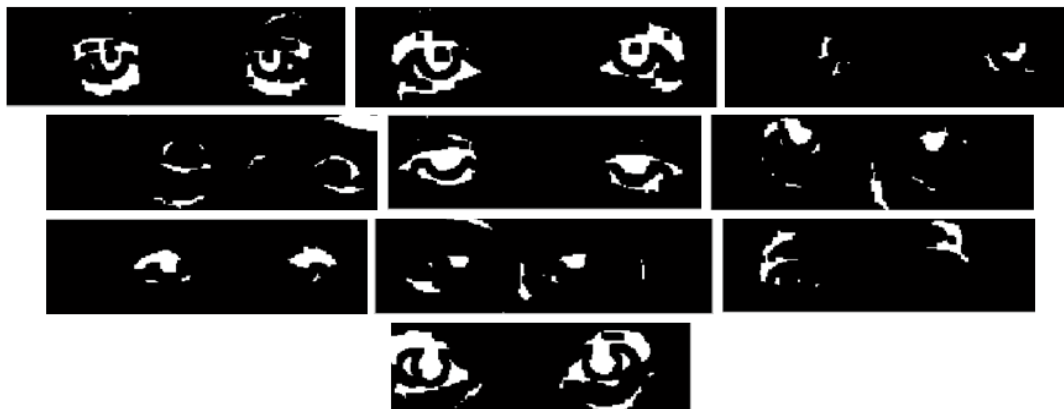


Figure 50: Participant eyes detected by bounding box method

Testing, as shown in Table 22, has been conducted on the same person, indoor, with background set as random (mostly colourful). It can be noticed that adding erosion to the processing stage of the images has improved the results of both eyes being detected.

Test #	Environment	Results	Reasons
Test 1	One fluorescent light bulb and noon sunlight, black scarf	Only left eye detected	Right eye too dark, lighting in room not equal
Test 2		Only left eye detected	Right eye too dark, lighting in room not equal
Changed threshold value of greyscale image			
Test 3		Both eyes detected	
Test 4		Both eyes detected	
Test 5		Only left eye detected	Right eye too dark, lighting in room not equal

Test 6	two fluorescent light bulbs, black scarf	Both eyes detected	
Test 7		Only left eye detected	
Test 8		Both eyes detected	
Test 9		Both eyes detected	
Test 10		Both eyes detected	
Test 11		Only left eye detected	Light in front of face
Erosion added			
Test 12	two fluorescent light bulbs, black scarf	Both eyes detected	
Test 13		Both eyes detected	
Test 14		Both eyes detected	
Test 15		Both eyes detected	
Test 16		Both eyes detected	
Test 17	very dim bulb light, black scarf	No eye detected	Camera not so advanced, poor quality images under dim light
Test 18	Noon sunlight through multi-coloured window, black scarf	Both eyes detected	
Test 19	Noon sunlight, black scarf	Both eyes detected	

Test 20	Dim lighting from fluorescent and incandescent bulbs and dim sunlight, black scarf	Both eyes detected	
Test 21	Only sunlight, colourful scarf	Both eyes detected	
Test 22		Both eyes detected	
Test 23		Both eyes detected	
Test 24		Both eyes detected	
Test 25		Both eyes detected	

Table 22: Testing results of bounding box eye calibration method

Line Laser

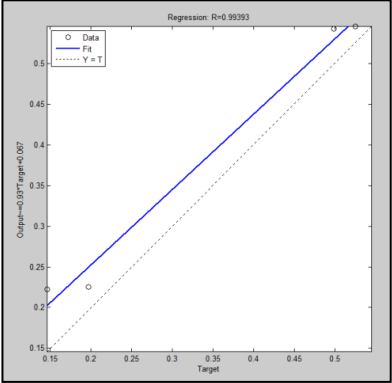
While this method requires an additional hardware set up (setting the line laser), it is advantageous in the fact that it requires only one frame. From this one frame, the face can be directly detected after processing of the image. Testing on one sample showed good results as long as the user is the nearest object to the screen and the laser is mounted in a proper horizontal position and distance from the user. As noticed, this method involves face detection prior to eye detection unlike previous method where eyes were detected directly without the need for face detection.

Data Fitting Neural Network, No. of Hidden Neurons = 20

After the network was trained and minimum MSE values were reached, the network was tested by randomly inserting images captured by the webcam. This is

to determine the accuracy by which the network can recognise faces and can correctly output the position of the area between the two eyes:

Desired Output			
0.2224	0.2256	0.5424	0.5456
Network Output			
0.273832	0.276808	0.571432	0.574408
MSE			
0.00224493			
Regression			
0.99393			



A good fit is noticed between the output and desired target, with a regression of 0.99393

Table 23: Testing results of data fitting neural network for face detection (20 hidden neurons)

As can be noticed from the Table 23, a close precision is achieved but not good enough to detect the desired area in the face. This is because when retrieving back the pixel position, a significant difference is noticed when comparing the output pixel position result and the desired pixel position result. This is illustrated in the calculation sections i) and ii) below:

i) Retrieving desired pixel position:

$$\begin{aligned} \text{position in input file: } & ([0.2224, 0.2256, 0.5424, 0.5456] \times 625 \\ & = [139, 141, 339, 341]) \end{aligned}$$

$$\text{position in picture: } [139, 141, 339, 341] \div 25 = [5.56, 5.64, 13.56, 13.64];$$

$$\begin{aligned} \text{if remainder} \neq 0, \text{row number} &= [5 + 1, 5 + 1, 13 + 1, 13 + 1] \\ &= [6\text{th}, 6\text{th}, 14\text{th}, 14\text{th}]; \end{aligned}$$

$$\text{if remainder} = 0, \text{row number} = [5\text{th}, 5\text{th}, 13\text{th}, 13\text{th}]$$

$$\begin{aligned} \text{column number: } [5, 5, 13, 13] \times 25 &= [125, 125, 325, 325] \\ &= [139 - 125, 141 - 125, 339 - 325, 341 - 325] \\ &= [14\text{th}, 16\text{th}, 14\text{th}, 16\text{th}] \end{aligned}$$

ii) Retrieving output pixel position:

$$\begin{aligned} \text{position in input file: } ([0.2738, 0.2768, 0.5714, 0.5744] \times 625 &= \\ [171, 173, 357, 359]) \end{aligned}$$

$$\text{position in picture: } [171, 173, 357, 359] \div 25 = [6.84, 6.92, 14.28, 14.36];$$

$$\begin{aligned} \text{if remainder} \neq 0, \text{row number} &= [6 + 1, 6 + 1, 14 + 1, 14 + 1] \\ &= [7\text{th}, 7\text{th}, 15\text{th}, 15\text{th}]; \end{aligned}$$

$$\text{if remainder} = 0, \text{row number} = [6\text{th}, 6\text{th}, 14\text{th}, 14\text{th}]$$

$$\begin{aligned} \text{column number: } [6, 6, 14, 14] \times 25 &= [150, 150, 350, 350] \\ &= [171 - 150, 173 - 150, 357 - 350, 359 - 350] \\ &= [21\text{th}, 23\text{rd}, 7\text{th}, 9\text{th}] \end{aligned}$$

Clearly the output pixel positions do not match accurately with the desired values, especially in case of the column number. However, row numbers are very close and show good precision to the desired values (desired row position is 6 and 14, output row position is 7 and 15).

Data Fitting Neural Network, No. of Hidden Neurons = 125

Similar test image to the one used in previous network (with 20 hidden neurons) has been tested on this network, and results tabulated in Table 24.

Desired Output			
0.2224	0.2256	0.5424	0.5456
Network Output			
0.165592	0.168248	0.431192	0.433848
MSE			
0.0119941			
Regression			
1			

Figure displays difference between desired output and measured output (result is not satisfactory)

Table 24: Testing results of data fitting neural network for face detection (125 hidden neurons)

Thus, performance has degraded when number of hidden neurons has increased to 125 units and over fitting is likely to happen.

Pattern Recognition Neural Network, No. of Hidden Neurons = 20

Random face image was inserted into network, and result is shown in Table 25.

		Confusion Matrix		
		1	2	3
Output Class	1	1 100%	0 0.0%	100% 0.0%
	2	0 0.0%	0 0.0%	NaN% NaN%
	3	100% 0.0%	NaN% NaN%	100% 0.0%
		Target Class		
		1	2	3

Image has been correctly recognised as a face and output was 1. A minimum mean square error (MSE) of 0.000000000844872 was achieved.

Table 25: Testing results of pattern recognition neural network for face detection (20 hidden neurons)

Thus, faces are well recognised. However, this network is a pattern recognition that is successful in recognising patterns and outputting 1, but unsuccessful in displaying the exact positions of the nose area which is required for skin colour extraction.

Data Fitting Neural Network, No. of Hidden Neurons = 20

This network is trained by data fitting, similar to the network trained previously. However, in this network, values of the training images were inserted in a column-wise manner, unlike the previous network where values of the training images were inserted row-wise. This has been done to test if networks learn differently depending on the pixel insertion order of the training images. A random face image was inserted into the network, with the following results in Table 26:

Desired Output			
331	339	381	389
Network Output			

310.3	320.7	375.3	385.7
-------	-------	-------	-------

Table 26: Testing results of data fitting neural network for face detection (20 hidden neurons, column-wise)

Thus, performance has not improved, which shows that insertion of data column-wise instead of row-wise does not guarantee better results.

Support Vector Machine, Polynomial Kernel

Next was to train support vector machines for face detection, observe the performance after training and compare the results with the previous methods. After the SVM is trained, the face detection ability of the trained SVM is tested. First step in testing is to capture and process face images, after which confidence level is calculated. This confidence level measures how much the captured image resembles a face as per the trained SVM. If the confidence level exceeds a set threshold, face is detected by the SVM.

Following Figure 51 shows the steps of the testing phase:

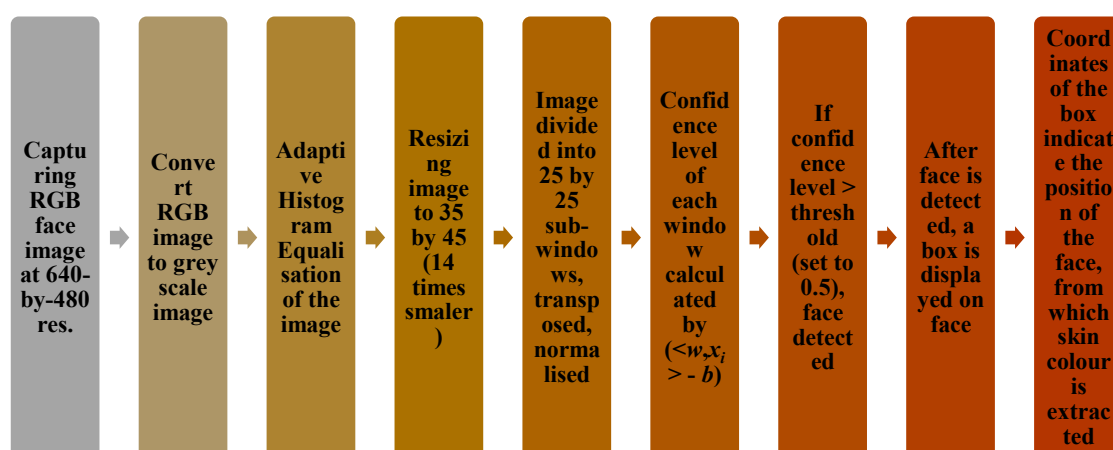


Figure 51: Steps of face detection by SVM

Image with box displayed over face is shown in Figure 52:

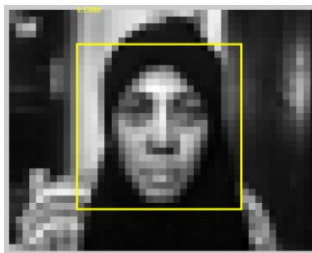


Figure 52: Face detected in yellow box by trained SVM

The confidence level, as displayed on the image, is 0.73888, which is above threshold set (= 0.5). The box coordinates are [10.5000 5.5000 34.5000 29.5000]. Dividing the box width by half gives the exact position in between the two eyes, as shown in Figure 53:

$$34.5 + 10.5 = 22.5$$



Figure 53: Middle of box of SVM detected face matches exactly the face centre

3.2 Testing of Tracking Phase

Combination 1: White and Black pixels in eyes

During testing of the calibration plus tracking stages, following are the testing results in Table 27. The testing was conducted indoor, since the system is designed to be used indoor only. Testing was conducted on six other people as well with different ages and gender. This is to test that the system can detect any human face regardless of variations in human faces, thus increasing the reliability of the system.

Test #	Environment	Results	Comments
Test 1	2 fluorescent bulbs and afternoon sunlight, black plain scarf	Eyes detected and tracked	
Test 2		Eyes detected and tracked	
Test 3		Eyes detected and tracked	
Test 4		Eyes detected and tracked	
Test 5	2 fluorescent bulbs and afternoon sunlight, black and white patterned scarf	Eyes detected and tracked	
Test 6		Error, eyes detected differently	Cropped area needs enlargement, face was too close to the laptop screen
Test 7	2 fluorescent bulbs and afternoon sunlight, colourful floral scarf	Eyes detected and tracked	
Test 8	Only afternoon sunlight	Eyes detected and tracked	
Test 9	Only evening sunlight	Eyes detected and tracked	
Small changes to program code, expanded the eye search area			
Test 10	Only evening sunlight	Eyes detected and tracked	Eye looking sideways away from camera, still eyes were tracked
Test 11	Very bright background of 2 fluorescent bulbs (bright glare due to bad camera alignment)	Eyes detected and tracked	
Other people testing			
Test 12	2 fluorescent bulbs and evening	Eyes detected and	Female

	sunlight	tracked	
Test 13	2 fluorescent bulbs and evening sunlight	Eyes detected and tracked	Male
Test 14	2 fluorescent bulbs	Eyes detected and tracked	Male
Test 15	2 fluorescent bulbs	Eyes detected and tracked	Male
Test 16	2 fluorescent bulbs	Eyes detected and tracked	Female (child)
Test 17	2 fluorescent bulbs	Eyes detected and tracked	Male (child) Had to remove his glasses for the system to recognise his eyes
Test 18	2 fluorescent bulbs	Eyes detected and tracked	Female
Test 19	2 fluorescent bulbs	Eyes detected and tracked	Female (removed glasses)
Test 20	2 fluorescent bulbs	Eyes detected and tracked	Female
Test 21	2 fluorescent bulbs	Eyes detected and tracked	Female

Table 27: Testing results of tracking combination 1 algorithm

Combination 2: Mathematical Face Measurements

The environment for testing was indoor, with incandescent lighting, and results displayed in Table 28.

Test #	Results	Comments
Test 1	Right eye tracked, left eye wrongly tracked	Black pupil of left eye not detected correctly
Test 2	Right eye tracked, left eye wrongly tracked	Black pupil of left eye not detected correctly
Test 3	Right eye and left eye tracked correctly	
Test 4	Right eye and left eye tracked correctly	
Test 5	Right eye and left eye tracked correctly	
Test 6	Eyes not tracked accurately	Colourful background was interfering with colour segmentation
Test 7	Right eye tracked, left eye wrongly tracked	Black pupil of left eye not detected correctly
Test 8	Right eye and left eye tracked correctly	
Test 9	Right eye tracked, left eye wrongly tracked	Black pupil of left eye not detected correctly
Test 10	Right eye and left eye tracked correctly	
Test 11	Left eye tracked, right eye wrongly tracked	Black pupil of right eye not detected correctly

Table 28: Testing results of tracking combination 2 algorithm

Combination 3: Eye Detection using SVM

The environment was indoor, with incandescent bulbs, and testing results displayed in Table 29.

Test #	Results	Participant
Test 1	Right eye and left eye tracked correctly	Female
Test 2	Right eye and left eye tracked correctly	Male
Test 3	Right eye and left eye tracked correctly	Male

Test 4	Right eye and left eye tracked correctly	Female/ Child
Test 5	Right eye and left eye tracked correctly	Female
Test 6	Right eye and left eye tracked correctly	Female
Test 7	Right eye and left eye tracked correctly	Female
Test 8	Right eye and left eye tracked correctly	Female
Test 9	Right eye and left eye tracked correctly	Female
Test 10	Right eye and left eye tracked correctly	Male
Test 11	Right eye and left eye tracked correctly	Male/ Child

Table 29: Testing results of tracking combination 3 algorithm

3.3 Testing of System

When comparing testing results of the different approaches in the calibration phase, best results were obtained with the trained SVM. Similarly, best results in the tracking phase were obtained by the combination 3 algorithm. Thus, a whole integrated system of detection and tracking has been built by combining face detection by SVM and face tracking by combination 3 algorithm. Testing of the whole integrated system was done indoors, under two florescent light bulbs, and following are the results in Table 30:

Test #	Results	Reason
Test 1	Key not displayed	Eyes detected correctly as 'centre'
Test 2	Key displayed	Eyes detected correctly as 'side'
Test 3	Key displayed	Eyes detected correctly as 'side'
Test 4	Key not displayed	Eyes detected wrongly as 'centre'
Test 5	Key not displayed	Eyes detected correctly as 'centre'
Test 6	Key not displayed	Eyes detected correctly as 'centre'
Test 7	Key displayed	Eyes detected correctly as 'side'

Test 8	Key not displayed	Eyes detected correctly as 'centre'
Test 9	Key displayed	Eyes detected correctly as 'side'
Test 10	Key displayed	Eyes detected wrongly as 'side'
Test 11	Key not displayed	Eyes detected correctly as 'centre'
Test 12	Key not displayed	Eyes detected correctly as 'centre'
Test 13	Key displayed	Eyes detected correctly as 'side'
Test 14	Key not displayed	Eyes detected correctly as 'centre'
Test 15	Key displayed	Eyes detected correctly as 'side'
Test 16	Key not displayed	Eyes detected correctly as 'centre'
Test 17	Key not displayed	Eyes detected correctly as 'centre'
Test 18	Key not displayed	Eyes detected correctly as 'centre'
Test 19	Key displayed	Eyes detected correctly as 'side'
Test 20	Key displayed	Eyes detected correctly as 'side'

Table 30: Testing results of integrated system

The Success Rate (SR) of the eye tracking keyboard can be calculated as:

$$\frac{\text{Number of correct display}}{\text{Total number of trials}} = \frac{18}{20} \times 100\% = 90\% \quad (20)$$

And the Error Rate (ER) is calculated as follows:

$$\frac{\text{Number of error}}{\text{Total number of trials}} = \frac{2}{20} \times 100\% = 10\% \quad (21)$$

The results of the testing of both calibration and tracking phases can be summarised as below:

- i) Multiple Frames Method in Calibration phase: Possibility of false eye detection is high

- ii) Bounding Box Method in Calibration phase: A SR of 96% (24 out of 25) was achieved.
- iii) Line Laser Method in calibration phase: Correct face detection but existence of hardware implementation requirement makes this method not preferable
- iv) Data Fitting (20 hidden neurons) Neural Network in calibration phase: MSE of 0.00224493, precise row number, non-precise column number
- v) Data Fitting (125 hidden neurons) Neural Network in calibration phase: MSE of 0.119941.
- vi) Pattern Recognition (20 hidden neurons) Neural Network in calibration phase: MSE of 0.844872×10^{-9} , correct face detection but non-precise pixel position extraction
- vii) Support Vector Machine in calibration phase: Exact face pixel position extracted.
- viii) Combination 1 algorithm in Tracking Phase: a SR of 95% (20 out of 21).
- ix) Combination 2 algorithm in Tracking Phase: a SR of 68% (7.5 out of 11).
- x) Combination 3 algorithm in Tracking Phase: a SR of 100% (11 out of 11).
- xi) SVM in Calibration Phase and Combination 3 algorithm in Tracking Phase: a SR of 90%.

Chapter 4: Discussion

There are a number of different approaches that exist for developing an eye tracking system, as can be seen in those developed by Praglin and Tan (2014), Liu et al (2010), Zia et al. (April 2014), Choi et al. (July 2011), Bengoechea et al. (September 2012), Fernandez et al. (April 2014), Majumder et al. (March 2011), Fosalau et al. (August 2011), Jiao et al (July 2014), and Abdel-Kader et al (2014). Therefore, in this research work, different combination methods were built, tested, and then compared to come up with the best approach.

4.1 Calibration Stage

In terms of calibration, nearly five different methods were compared, and it was found that each method has its own pros and cons. The comparative results are shown in Table 31. The different calibration approaches were compared based on some features as shown in Table 32.

Based on obtained testing results, it was found that each method has some advantages and disadvantages over the other. However, the final choice was to limit the comparison to only ANN and SVM. This is because of their practicality and automatic face detection after training is completed. While comparing these two, it was noticed that the performance of SVM is superior to that of ANN. Accurate results were reached in case of SVM when same training samples were used for training both SVM and ANN. However, it was also observed that the performance of ANN can be improved.

Multiple Frames (1)	<p>Pros: No hardware set up is required. It is easy method for the user since user is only required to stare at screen or camera and blink naturally. The user might stare at any other location as well as long as the head is not tilted to a big degree. Thus, it is practical to the user (here, focus is on disabled person). If best image is picked out of all other images, accurate position of the eyes can be extracted.</p> <p>Cons: System might fail to pick out the best image showing exact eyes. This is because other white pixels might be left out after image processing and interpreted as the eyes. In this case, wrong eyes will be detected. Also, this method involves many frames to loop over to select the best frame, and consumes more system memory.</p>
Bounding Box (2)	<p>Pros: No hardware is required, only software. It is a very simple and straight forward approach that captures only two frames, one of closed eyes and the other of opened eyes, from which the eyes position is extracted. Also, eyes position is extracted accurately with no minimum errors on testing.</p> <p>Cons: The user needs to stare at camera; slight movement might produce wrong eyes position.</p>
Line Laser (3)	<p>Pros: This method is so fast in implementation. Only one frame is captured to detect the face. Also, user can be positioned at any angle and the head can be freely tilted since this method depends on concept of distance from the line laser and reflection from the closest object. There are no complications of capturing multiple frames.</p> <p>Cons: This method requires software and hardware setup, thus not practical in terms of set up. The line laser needs to be horizontally positioned or the face will be detected but measurements will be wrong. Under extremely bright environments, the line laser might not be completely visible. Other objects placed close to the screen (and hence laser) might be wrongly detected as the face; face is detected prior to the eyes unlike previous methods in which eyes were detected directly - line laser cannot be directed directly on the user's eyes.</p>
Artificial Neural Networks	<p>Pros: While other approaches require calibration whenever a new user is introduced, this method involves training the network only once. After training, the network can be used for any new user without the need to re-train. Thus, it is practical and time saving after</p>

(4)	<p>training.</p> <ul style="list-style-type: none"> • Cons: Training the network, choosing the network architecture and training parameters is more complicated and time consuming compared to the processing in previous methods. With 34 face samples for training and 1030 training iterations, the output produced was accurate enough to locate the exact eyes position on the face.
Support Vector Machines (5)	<ul style="list-style-type: none"> • Pros: training such machines is faster than training neural networks. Once trained and margin is established between positive and negative samples, testing an image is fast where only confidence level is calculated. Accurate results were noticed even with small number of training samples. • Cons: While accurate results were noticed, some images are still mistaken as faces due to similarity in pixel values (colour pattern).

Table 31: Pros and cons of the different calibration methods

This is because small values of mean squared error (MSE) were reached, and the network produced accurate results in terms of the row number. Inaccurate results were only noticed with regards to the column number desired. It is believed that the performance of ANN can be improved by increasing the number of training face samples, or by varying the training samples more. For instance, varying the lighting conditions for each training face sample might produce good results (with precise column position) for the ANN. Thus, SVM was chosen as the optimum approach in the calibration phase.

Feature	Calibration Methods				
	1	2	3	4	5
Speed	15.35628 seconds	19.674946 seconds	2.536435 seconds	1030 iterations of training (5 secs)	No iterations, weight matrix calculated (0.2 secs)
Software req.	Yes	Yes	Yes	Yes	Yes
Hardware req.	No	No	Yes	No	No
No. of Frames	Multiple (15)	Two	One	34 training images	68 (both face and non-face)
No. of times running code	For every new user	For every new user	For every new user	Calculate weight matrix only once during training	Develop margin only once during training
Accuracy of Results	Accurate when best frame selected	Very accurate	Accurate if user is closest object to screen	Row number accurate; Column number not very accurate	Accurate in majority of samples
Training Accuracy	-	-	-	99.4613% (training samples) 81.5132% (validation samples)	100% (training samples) 87.5 % (validation samples)

Table 32: Comparison between the different calibration methods

4.2. Tracking Stage

The three tracking methods discussed in the previous chapter vary in terms of the main technique. While combination 1 depends on the search for black and white pixels of the eyes and implements bounding box as the calibration method, combination 2 works on studying face measurements and combination 3 works on trained eye SVM as well as black pixels of the eye pupil. A couple of pros and cons exist with each method, as discussed in Table 33:

Combination 1	Combination 2	Combination 3
<p>Pros: Easy and fast code that only searches for cornea and pupil of the eye after histogram equalisation.</p> <p>Cons: Depending only on black and white pixels of the eye for tracking is not practical since lighting conditions can hinder correct results.</p>	<p>Pros: Limiting the search area for tracking eyes tends to produce more accurate results.</p> <p>Cons: This method depends on the accuracy of face detection by colour segmentation. Any mistakes in the face measurements can result in wrong tracking of the eyes.</p>	<p>Pros: Depends on both black pixels of eye pupil as well as the results of a trained SVM to recognise eyes. Also checks that both eyes are aligned together horizontally; thus correct results are guaranteed</p> <p>Cons: The code turned out to be longer with more processing steps.</p>

Table 33: Pros and cons of the different tracking methods

In order to come up with methods for eye tracking a number of techniques were tried but were found to fail in tracking the eyes. Some of these techniques are:

Morphological operators: These operators attempt to detect eyes with the help of their unique shape. There is no doubt about the efficiency of morphological operators for eye detection, as implemented in the project of DHRUW (2009). However, it was found that edge detection to detect the ellipse shape of the eyes failed in this

research work. Reasons were that the camera is of low resolution, and the system is built to run indoors under room lightings. This lighting is usually dim, and shadows tend to hinder the shape of the eyes in the captured images.

Eye RGB pixel values: An attempt was made to design a system to track eyes based on exact RGB values of the eyes obtained during calibration phase. This method is similar to skin colour segmentation, but applied to eyes. It was thought that this method will be suitable since people have different colours of the pupil. However, the search for the RGB values of the eyes was found to be unsuccessful since the program reads the exact RGB pixel values. There exists blocks on a human face where the mean and standard deviation of its pixels are the same as that of the eyes, even if the block colours look different to an ordinary human eye. This is illustrated in Table 34:

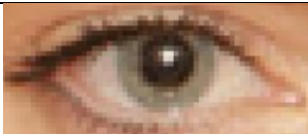
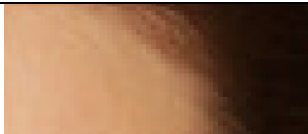
Intended eye to track based on the mean and standard deviation of the RGB pixels	Other face block with same mean and standard deviation like Eye RGB values
	

Table 34: Failure of eye colour segmentation in eye tracking

White cornea of the Eyes: Two features unique in any eye are the black pupil and white cornea. While the former was successful in detecting the eyes, the latter produced unsuccessful results in some cases. It was found that the eye cornea pixels are not the highest in terms of value. There are pixels in the human face with larger numbers of pixel values (hence brighter, closer to 255) and are falsely detected by the system as the eye cornea. This is because the computer reads the pixel values,

unlike the human eyes that depend on the appearance. Also, the lighting conditions can make the cheeks or nose area containing brighter pixel values than the eyes. The eyes area is always hollow and shaded. As such, the search for the white cornea pixels has been discarded from the tracking system.

With respect to the tracking combinations that were built, the combination 3 technique was finally selected for the system. This is because this technique was designed after studying the drawbacks of the previous two combinations and putting solutions for each drawback into the combination 3. As such, all failures of eye tracking in combination 2 are solved in combination 3. Also, combination has if/else conditions to further check if eyes are tracked correctly. Overall, the testing on combination 3 produced the best results as below, in Figures 54 and 55.

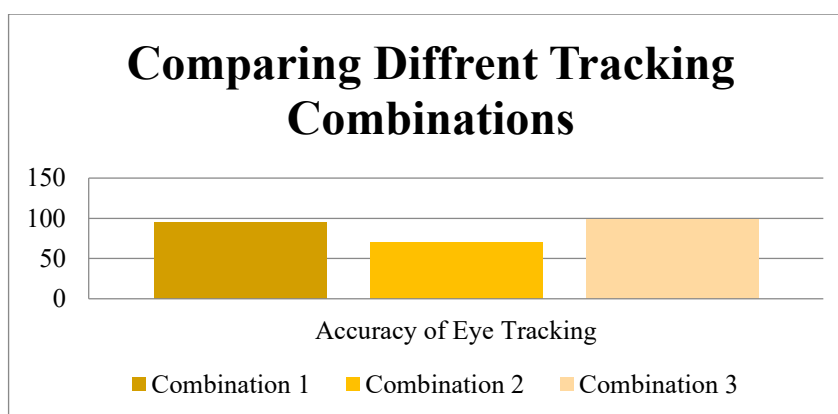


Figure 54: Comparison of Accuracy Rate of the different tracking algorithms

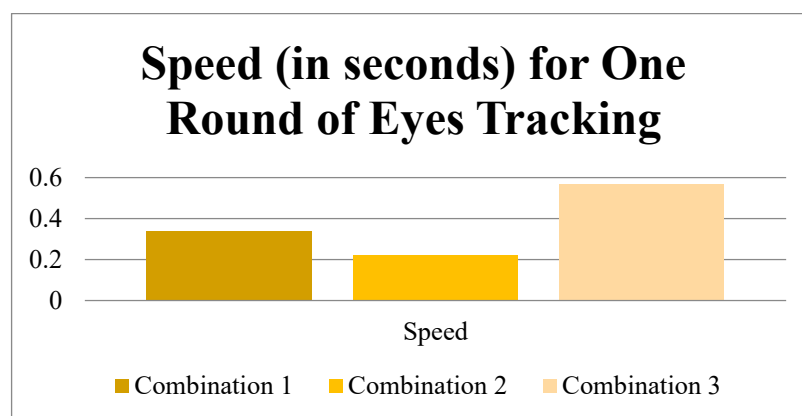


Figure 55: Comparison of Speed of the different tracking algorithms

The combination 3 produced best results, which was expected prior to testing since it was built to overcome problems in previous codes. Also, combination 3 involves tracking by a trained SVM to detect eyes. While both the SR and speed are important, the SR has more priority than the speed.

The presence of an object with same colour as the skin degrades the system performance and might cause inaccurate tracking of the eyes. Therefore, the background is preferred to be plain with a colour different than that of the user's skin colour. This is because skin colour segmentation is the main method of tracking in this research project. Face detection happens prior to eye detection. Nevertheless, the program works properly in other cases, and has been tried on both female and male, including children. Also, face was detected regardless of the eyes being closed or opened. This is because the SVM designed to detect face during calibration stage was trained with samples of both closed eyes and opened eyes. In terms of face orientation, the system was designed to detect eyes that are aligned horizontally. However, a small deviation of the head along the horizontal line will not affect the system performance. Also, this degree can be controlled in the program code to meet the needs of the user.

Chapter 5: Conclusion

5.1 Managerial Implications

Eye tracking can be developed in a number of ways. When building an eye tracking system, the challenge is to choose the method of implementation. This research work explores a number of techniques, like colour segmentation, morphological operations, artificial neural networks, support vector machines and so on. It combines a number of methods together to come up with an efficient eye tracking system. The aim, here, was to develop a working system using simple methods and limited hardware. The advantage of the method proposed in this research work is that it uses a normal webcam that is affordable by all people. As such, this system could be software implemented and installed on any device.

Alternative ways also exist to build such eye tracking system. Using a depth sensor camera or a more advanced camera will make the tracking code easier and shorter. This is because the working on the system will depend on both the hardware and software power. In this research work, it is mostly dependent on the software efficiency. But if a depth sensor camera was used, for instance, it will work by regarding the closest object as the user and no more complications in the code are required. However, using an advanced camera implies that users need to buy a hardware device (since a depth camera is not as common as a webcam). Thus, this implies that there is a trade-off between simple software with hardware requirement and complicated software with no hardware requirements. In this research work, the choice was a complicated software with a webcam that is affordable by all people.

In future work, hardware implementation can help improve the working of this project. Chips might enhance the speed of the code. Parallel computation can also be applied. The ability of the SVM to detect faces and eyes can be enhanced by choosing more efficient training samples. For instance, increasing the number of samples, and making it more varied by varying the lighting conditions and including people with larger age span and different ethnic backgrounds. The program can also be designed to allow more freedom in terms of head orientation and face expressions.

Eye Tracking and gesture recognition are very broad topics. When tracking the eyes, many works focus on tracking the reflection on the eye pupil through the use of a reflected IR light, instead of movement of the eye pupil from side to side. Every approach has some advantages and disadvantages.

One of the drawbacks of skin colour segmentation techniques is the presence of similar skin colour object in the background. This limitation also applies to this project. While this research project implements skin colour segmentation and a couple of other operations to form a customised algorithm, other works prefer using well-known algorithms like Viola-Jones and Cascaded Classifiers for face and eye detection. Also, in skin colour segmentation, some researchers prefer working in the normal RGB colour space, as is the case in this research. Other researchers prefer working in LAB, HSV or HSI colour spaces. No colour space has proved dominance over the other methods in terms of performance. In fact, it depends on the code designer and the priorities set and objectives to achieve.

The keyboard designed was made as simple as possible. However, more buttons can be added. Also, common used words can be included, just like in AAC (Augmented and Alternative Communication). Since in the Middle Easter region,

people tend to be bilingual, and speak English along with their native language, the keyboard can be customised accordingly and allow switching between two/multiple languages.

5.2 Research Implications

With advancements in imaging and video (Memon, 2006), integration of modern technologies (Memon and Khoja, 2009; Memon and Khoja, 2010) and improvement in human life standard, many integrated technologies have surfaced to help solve social problems. One such problem is the difficulty faced by disabled people in leading a normal life. Projects, such as the proposed research work, help putting solutions to such problems. The developed keyboard can be easily used by people with physical disabilities and paralysis to type without any physical interaction. With the present ignorance in the Middle Eastern region towards the needs of special needs people and lack of facilities offered to them, this research work serves a good purpose in the region.

Furthermore, this proposed project suggests a unique and novel method of combining a number of techniques together to come up with eye detection and tracking system. This method was compared with other works, and proved its efficiency. The comparison has been made with the eye detection and tracking algorithms developed by Praglin and Tan (2014) and Zia et al. (April, 2014), as illustrated in Table 35:

Comparison with other works		
	Accuracy	Number of samples
MAP training, SIFT, RANSAC, homography (Praglin and Tan, 2014)	94% single eye detection, 50% both eye detection	50
Skin colour segmentation, Circular Hough Transform (Zia et al., April 2014)	77.78% single eye detection, 66.67% both eye detection	9
Proposed algorithm	90% double eye detection	20

Table 35: Comparison with other research works

In the work by Praglin and Tan (2014), images are of resolution 1944×1296, with a single core of i7. The proposed algorithm produces good results despite the low resolution of 640×480 of captured images. However, the processor is an i7 dual core. With both the proposed algorithm and that of Zia et al. (2014) implementing skin colour segmentation for face detection, higher accuracy rate is achieved in the former.

The system allows the user to be seated anywhere between 60 to 180 cm away from the computer screen. In terms of head orientation, the skin colour segmentation will detect faces tilted by any angle. However, the SVM was trained with eye samples looking straight into the camera. As such, a huge angle tilt of the user's head will mostly lead to false tracking by the camera, since the confidence level will not exceed the threshold set for the SVM to detect eyes. A small angle head tilt of an angle of 0-15 degrees (along X-Y axis, as in Figure 56) is possible. In the future, the SVM could be trained to detect 45 degree head tilt or even a tilt along the z axis

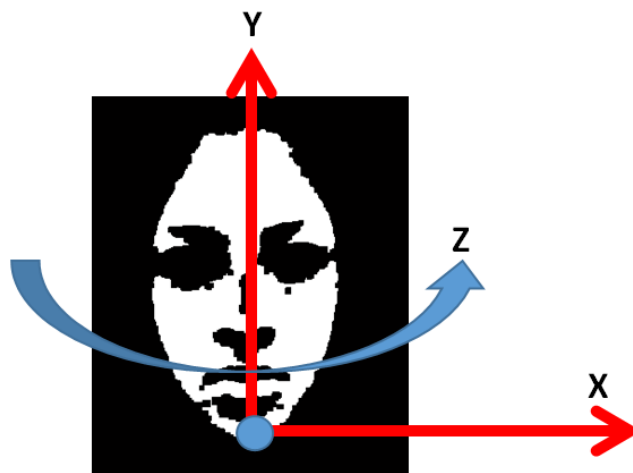


Figure 56: Head Tilt Graph along the XYZ axis

This opens doors for further improvements to the code, as discussed in the previous section, as well as real life implementation. Specifically, a couple of methods were implemented in this project. While some methods produced good results, other methods produced unsatisfactory results. It was found that SVM and ANN produce good results in terms of face and eye detection, thus confirming the efficiency of machine learning in solving such issues. However, their performance depends largely on the training samples. SVM needs both positive and negative training samples. However, the performance of the SVM was slightly more satisfactory. The skin colour segmentation was also found to be successful under the condition that the background does not contain colours similar to the user's skin. Thus, it's better to implement skin colour segmentation under the conditions that the background is plain. Furthermore, image processing operations such as histogram equalisation and erosion were found to greatly help in tracking. The former is useful since it improves the contrast of colours in the image. The latter helps improve the quality of the black and white picture of the face extracted after skin colour segmentation. Overall, the appearance-based algorithm has proved efficient in eye and face tracking.

Other methods were implemented, but unexpectedly, didn't produce satisfactory results. For instance, the elliptical shape of eyes can be exploited for eye detection. However, on application of edge detection, it was found that the algorithm failed to detect the eyes. This can be due to the low resolution of pictures captured by the webcam. Also, the pictures are captured when the user is seated between 60 and 180 cm (the comfort zone as described by Apple Inc. (2015)). As such, the eyes do not appear clearly in captured pictures. In the project of DHRUW (2009), edge detection has been implemented for both the elliptical shape of the eyes and the circular shape of the eye pupil, only under the condition that the person is close to the camera. This supports the previous explanation. Other methods that didn't produce satisfactory results on implementation are the RGB eye search and the white pixels cornea search. This can be explained by the fact that the program reads pixel values in a way different than that of a human eye. While the eye cornea might seem to be the brightest on a human face, in terms of pixel values, other face parts might have brighter values. Also, it was found that the surrounding illumination might affect the results. As such, some researchers prefer working on HSV/HSI colour spaces under the assumption that lighting does not affect the colours of the image.

Developing the eye tracking keyboard was done in sequential steps, starting with the calibration stage, following tracking and classification. One method was tested at a time. At the end of each stage, the different methods were compared and that method with best testing results was selected. As such, SVM was selected for the calibration stage, while a combination of methods were implemented in tracking.

Bibliography

- Abdel-Kader, R. F., Atta, R., & El-Shakhabe, S. (2014). An efficient eye detection and tracking system based on particle swarm optimization and adaptive block-matching search algorithm. *Engineering Applications of Artificial Intelligence*, 31, 90-100.
- Akl, A., Feng, C., & Valaee, S. (2011). A novel accelerometer-based gesture recognition system. *Signal Processing, IEEE Transactions on*, 59(12), 6197-6205.
- Al-Gain, S. I., & Al-Abdulwahab, S. S. (2002). Issues and obstacles in disability research in Saudi Arabia. *Asia Pacific Disability Rehabilitation Journal*, 13(1), 45-49.
- Al-Jadid, M. S. (2013). Disability in Saudi Arabia. *Saudi medical journal*, 34(5), 453-460.
- AlKassim, Z., "Virtual laser keyboards: A giant leap towards human-computer interaction," *Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on* , vol., no., pp.1,5, 18-20 Dec. 2012
- Alon, J., Athitsos, V., Yuan, Q., & Sclaroff, S. (2009). A unified framework for gesture recognition and spatiotemporal gesture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9), 1685-1699.
- ALS Association (2015). Retrieved online July 2015 from <http://www.alsa.org/about-als/what-is-als.html>
- Apple Inc. (2015). Retrieved May 2015 from <https://www.apple.com/>
- Arora, K., Suri, S., Arora, D., & Pandey, V. (2014). Gesture Recognition Using Artificial Neural Network. *JCSE International Journal of Computer Sciences and Engineering, Volume-2, Issue-4, E-ISSN: 2347-2693*
- Bengoechea, J. J., Villanueva, A., & Cabeza, R. (2012, September). Hybrid eye detection algorithm for outdoor environments. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 685-688). ACM.
- Celebi, S., Aydin, A. S., Temiz, T. T., & Arici, T. (2013). Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping. In *VISAPP (1)* (pp. 620-625).
- Chen, M., AlRegib, G., & Juang, B. H. (2013). Feature Processing and Modeling for 6D Motion Gesture Recognition. *Multimedia, IEEE Transactions on*, 15(3), 561-571.

- Choi, I., Han, S., & Kim, D. (2011, July). Eye detection and eye blink detection using adaboost learning and grouping. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on* (pp. 1-4). IEEE.
- Cincinnati Children's Hospital Medical Center (2011, December 2nd). Communicating with Partner Assisted Scanning [video file]. Retrieved from <http://www.youtube.com/watch?v=nGpSXQKrmR4>
- Creative Crash (2010, June 15th). Realistic Muscled Male Body (textured) 3D Model [Photograph]. Retrieved online 2014 from <http://www.creativecrash.com/3d-model/realistic-muscled-male-body-textured>
- Daniel, P., Cristian, F., Avila, M., & Felix, M. (2011, October). Algorithm for face and eye detection using colour segmentation and invariant features. In *Telecommunications and Signal Processing (TSP), 2011 34th* (pp. 564-569).
- DHRUW, K. K. (2009). *A. EYE DETECTION USING VARIANTS OF HOUGH TRANSFORM B. OFF-LINE SIGNATURE VERIFICATION* (Doctoral dissertation, National Institute of Technology Rourkela).
- Fernandez, M., Christina, D., Gob, K. J. E., Leonidas, A. R. M., Ravara, R. J. J., Bandala, A. A., & Dadios, E. P. (2014, April). Simultaneous face detection and recognition using Viola-Jones Algorithm and Artificial Neural Networks for identity verification. In *Region 10 Symposium, 2014 IEEE* (pp. 672-676). IEEE.
- Frolova, D., Stern, H., & Berman, S. (2013). Most probable longest common subsequence for recognition of gesture character input. *Cybernetics, IEEE Transactions on*, 43(3), 871-880.
- Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.
- Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4), 18-28.
- Holland, E. (2008). Marquardt's Phi mask: Pitfalls of relying on fashion models and the golden ratio to describe a beautiful face. *Aesthetic plastic surgery*, 32(2), 200-208.
- Human Diseases and Conditions. Paralysis [Photograph]. Retrieved online 2014 from <http://www.humanillnesses.com/original/Pan-Pre/Paralysis.html>

- INRIA Visual Recognition and Machine Learning Summer School (July, 2012). France. Retrieved online 2015 from <http://www.di.ens.fr/willow/events/cvml2012/>
- Jiao, Y., Peng, Y., Lu, B. L., Chen, X., Chen, S., & Wang, C. (2014, July). Recognizing slow eye movement for driver fatigue detection with machine learning approach. In *Neural Networks (IJCNN), 2014 International Joint Conference on* (pp. 4035-4041). IEEE.
- Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1), 103-108.
- Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (Vol. 1, pp. I-900). IEEE.
- Litwiller, D. (2001). CCD vs. CMOS. *Photonics Spectra*, 35(1), 154-158.
- Liu, A., Li, Z., Wang, L., & Zhao, Y. (2010, September). A practical driver fatigue detection algorithm based on eye state. In *Microelectronics and Electronics (PrimeAsia), 2010 Asia Pacific Conference on Postgraduate Research in* (pp. 235-238). IEEE.
- Majumder, A., Behera, L., & Subramanian, V. K. (2011, March). Automatic and robust detection of facial features in frontal face images. In *Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on* (pp. 331-336). IEEE.
- Memon, Q., and Khan, S. (1998), "Artificial Neural Network Approach to Camera Calibration and 3-D World Reconstruction for stereovision", *International Workshop on Recent Advances in Computer Vision*, pp. 35-40.
- Memon, Q., and Khan, S. (2001), "Camera Calibration and 3-D World Reconstruction for Stereo-vision using Neural Networks", *International Journal of Systems Sciences*, 32(9), pp. 1155-1159.
- Memon, Q. (2003), "Crime Investigation and Analysis using Neural Networks", *Proceedings of IEEE National Multitopic Conference*, Islamabad, December 9-10.
- Memon, Q., Laghari, S. (2006), "Wear Particle Analysis Based on Self Organizing Clusters", *International Journal of Robotics and Automation*, 21(4), pp. 282-287.

- Memon, Q. (2006), "A New Approach to Video Security over Networks", *International Journal of Computer Applications in Technology*, 25(1), pp. 72-83.
- Memon, Q. (2007), "Relation Based Clustering of Wear Particle Measurements for Industrial Automation", *International Journal of Automation and Control*, 1(2-3), pp. 207-219.
- Memon, Q. (2008), "Intelligent Control of Air-Traffic landing Sequences", *International Journal of Modeling and Simulation*, 28(1), pp. 4489, 2008.
- Memon, Q., Khoja, S. (2009), "Academic Program Administration via Semantic Web – A Case Study", *Proceedings of International Conference on Electrical, Computer, and Systems Science and Engineering*, Dubai, 37, pp. 695-698.
- Memon, Q., Khoja, S. (2010), "Semantic Web for Program Administration", *International Journal of Emerging Technologies in Learning*, 5(4).
- Meunier, F., ing, & Ph. D. (2009). *On the automatic implementation of the eye involuntary reflexes measurements involved in the detection of human liveness and impaired faculties*. INTECH Open Access Publisher.
- Mistry, P., & Maes, P. (2009, December). SixthSense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches* (p. 11). ACM.
- Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3), 311-324.
- Mori, A., Uchida, S., Kurazume, R., Taniguchi, R. I., & Hasegawa, T. (2010, November). Automatic construction of gesture network for gesture recognition. In *TENCON 2010-2010 IEEE Region 10 Conference* (pp. 923-928). IEEE.
- Nguyen, T. N., Huynh, H. H., & Meunier, J. (2013). Static Hand Gesture Recognition Using Artificial Neural Network. *Journal of Image and Graphics*, 1(1).
- Paralysis, National Health System (2014, August 28th). Retrieved online 2014 from <http://www.nhs.uk/Conditions/paralysis/Pages/Introduction.aspx>
- Pavlovic, V., Sharma, R. & Huang, T. (1997), "Visual interpretation of hand gestures for human-computer interaction: A review", *IEEE Trans. Pattern Analysis and Machine Intelligence.*, July, 1997. Vol. 19(7), pp. 677 -695.
- Praglin, M., & Tan, B. (2014). Eye Detection and Gaze Estimation. *Eye*, 1.

- Ramirez-Cortes, J., Gomez-Gil, P., Sanchez-Perez, G., & Prieto-Castro, C. "Shape-based hand recognition approach using the morphological pattern spectrum", *J. Electron. Imaging*. 18(1), 013012 (March 19, 2009).
- Razvan (2011). Gesture Recognition [Photograph]. Retrieved online 2014 from http://en.wikipedia.org/wiki/Gesture_recognition
- Reeve, C. (2002). One Degree of Separation: Paralysis and Spinal Cord Injuries in the United States. Christopher and Dana Reeve Foundation. Retrieved online from <http://www.christopherreeve.org/atf/cf/%7B3d83418f-b967-4c18-8ada-adc2e5355071%7D/8112REPTFINAL.PDF>
- Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *Multimedia, IEEE Transactions on*, 15(5), 1110-1120.
- Shawn (2010). Kinect, Great Start. Will it Continue? [Photograph]. Retrieved online 2014 from <http://mygggo.com/tag/kinect/>
- Shutterstock. com (2015) [Photograph]. Retrieved online May 2015 from <http://www.shutterstock.com/video/clip-2713739-stock-footage-surgical-operations-on-the-human-eye.html>
- Stergiopoulou, E., & Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8), 1141-1158.
- Suarez, J., & Murphy, R. R. (2012, September). Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE* (pp. 411-417). IEEE.
- Zhou, S., Fei, F., Zhang, G., Mai, J., Liu, Y., Liou, J., & Li, W. (2014). 2D Human Gesture Tracking and Recognition by the Fusion of MEMS Inertial and Vision Sensors. *IEEE SENSORS JOURNAL, VOL. 14, NO. 4*, 1160-1170.
- Zia, M. A., Ansari, U., Jamil, M., Gillani, O., & Ayaz, Y. (2014, April). Face and eye detection in images using skin color segmentation and circular hough transform. In *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on* (pp. 211-213). IEEE.

List of Publications

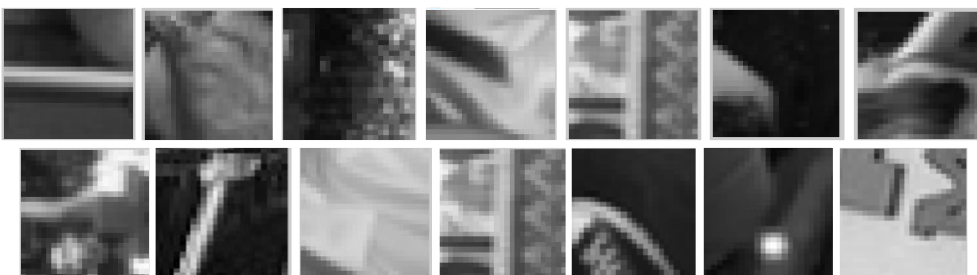
Alkassim, Z and Memon, Q (2015, August). Experimental Analysis of Camera Calibration Techniques used for Eye Tracking, *ICOCI 2015*, Istanbul, Turkey. (Best Paper Award)

Appendix

Sample of positive face images for machine learning:



Sample of negative face images for machine learning (captured randomly or taken from INRIA Visual Recognition and Machine Learning Summer School, 2012):



Sample of face detection and extraction by skin colour segmentation:



Sample of positive eye images for machine learning (captured or from internet):



Sample of negative eye images for machine learning (captured or taken from INRIA Visual Recognition and Machine Learning Summer School, 2012):



Sample of pupil position detection:

