**EYE GAZE BASED INTERACTION FOR PHYSICALLY DISABLE PATIENTS**

**ABDUL MOEEZ KHAN**

**MALIK MUHAMMAD USMAN**

**MUHAMMAD UMAIR**

**MUHAMMAD HAMMAD RASHID**

**RAHIM AHMED**

**A project report submitted in partial fulfilment of the**

**Requirements for the award of the degree of**

**Bachelor of Computer Science**

**Department of Computer Science**

**Bahria University, Karachi Campus**

**Tuesday, 11 December 2018**

**DECLERATION**

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reg No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date : Monday, 10 December 2018

**APPROVAL FOR SUBMISSION**

I/We certify that this project report entitled **EYE GAZE BASED INTERACTION FOR PHYSICALLY DISABLE PATIENT** was prepared by **ABDUL MOEEZ KHAN, MALIK MUHAMMAD USMAN, MUHAMMAD UMAIR, MUHAMMAD HAMMAD RASHID, RAHIM AHMED** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Computer Science (Honours) at Bahria University.

Approved by,

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor : MR AZMAT

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The copyright of this report belongs to the author under the terms of the copyright Ordinance 1962 as qualified by Intellectual Property Policy of Bahria University. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2018, ABDUL MOEEZ KHAN, MALIK MUHAMMAD USMAN, MUHAMMAD UMAIR, MUHAMMAD HAMMAD RASHID, RAHIM AHMED All right reserved.

**ACKNOWLEDGEMENTS**

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, MR AZMAT for his invaluable advice, guidance and hisenormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

**EYE GAZE BASED INTERACTION FOR PHYSICALLY DISABLE PEOPLE**

ABSTRACT

Human-computer interaction (HCI) has grown exponentially in the last decade. The arrival of tablets and cell phones that allow touch control has been warmly praised. Researchers have also explored the potential of eye sight as a possible means of interaction. To help disabled patients, some commercial solutions have already been launched, but they are still expensive and provide limited use capacity. This project encourages the introduction of a real-time low-cost system for human-computer interaction based on perception. Eye tracking is an important area in the medical sector, especially in psychiatry to give an idea of patients with mental disorders. Today, the presence of portable devices with powerful devices can provide an opportunity to investigate whether the use of these devices can be traced without using any additional devices. This project will attempt to explore the possibility of tracking center of the pupil using computers. To achieve this, HaaR eye tracking algorithms are applied, comparing templates and revealing the eye using image gradients. The application was also applied as a background service and as an independent activity to investigate the performance and ease of use of these two methods. HaaR algorithms, pattern matching, and eye detection using chromatography algorithms show promising results.

Table OF CONTENTS

[ABSTRACT vii](#_Toc532137423)

[Table OF CONTENTS viii](#_Toc532137424)

[LIST OF FIGURES xii](#_Toc532137425)

[LIST OF SYMBOLS / ABBREVIATIONS xiii](#_Toc532137426)

[LIST OF APPENDICES xiv](#_Toc532137427)

**CHAPTER**

[INTRODUCTION 15](#_Toc532137384)

[Background 15](#_Toc532137385)

[Problem statement 15](#_Toc532137386)

[LITERATURE REVIEW 17](#_Toc532137387)

[Existing Solution 17](#_Toc532137388)

[Type of data obtained 17](#_Toc532137389)

[An important reference point 18](#_Toc532137390)

[Three-dimensional direction of sight 18](#_Toc532137391)

[Methods for measuring the position of the eye 20](#_Toc532137392)

[Electro-Eye tracking 20](#_Toc532137393)

[Technician using contact lenses 20](#_Toc532137394)

[Optical techniques based on video recording 20](#_Toc532137395)

[Without using infrared light 22](#_Toc532137396)

[2.3. Review of existing solutions 23](#_Toc532137397)

[Theoretical basics 23](#_Toc532137398)

[Visual system 23](#_Toc532137399)

[Description of the visual system 24](#_Toc532137400)

[Eye movement 25](#_Toc532137401)

[Optical flow 25](#_Toc532137402)

[DESIGN AND METHODOLOGY 27](#_Toc532137403)

[Tracking Algorithms 28](#_Toc532137404)

[Tracking head position 28](#_Toc532137405)

[Active Appearance Model 29](#_Toc532137406)

[The algorithm of face and eye detection 30](#_Toc532137407)

[Haar Feature-based Cascade Classification 32](#_Toc532137408)

[Template Matching Algorithm 33](#_Toc532137409)

[Timm’s and Barth’s Algorithm Using Image Gradients 34](#_Toc532137410)

[Determination of tracking characteristics 36](#_Toc532137411)

[Algorithm for determining traits for tracking 38](#_Toc532137412)

[Use of reference frames 38](#_Toc532137413)

[Implementation 40](#_Toc532137414)

[4.1 Face detection 40](#_Toc532137415)

[4.2 Eye-region extraction 41](#_Toc532137416)

[4.3 Eye-region extraction 42](#_Toc532137417)

[4.4 Eye-blink classification 42](#_Toc532137418)

[4.5 Code 42](#_Toc532137419)

[RESULTS AND DISCUSSIONS 52](#_Toc532137420)

[CONCLUSION AND RECOMMENDATIONS 54](#_Toc532137421)

[REFERENCES. 56](#_Toc532137422)

LIST OF FIGURES

**FIGURE TITLE PAGE**

[Figure 1 JAZZ-novo measuring system 17](#_Toc517833933)

[Figure 2 Eye test results represented by the heat map 18](#_Toc517833934)

[Figure 3 3D Video system 18](#_Toc517833935)

[Figure 4 (a) Bright source effect caused by the source 20](#_Toc517833936)

[Figure 5 Four types of Purkinje images 21](#_Toc517833937)

[Figure 6 Construction of the eye 23](#_Toc517833938)

[Figure 7 : System diagram 26](#_Toc517833939)

[Figure 8 Face and eye detection 30](#_Toc517833940)

[Figure 9 Haar features 31](#_Toc517833941)

[Figure 10 template matching method 32](#_Toc517833942)

[Figure 11 slide the template for matching 32](#_Toc517833943)

LIST OF SYMBOLS / ABBREVIATIONS

*cp* specific heat capacity, J/(kg⋅K)

*h* height, m

*Kd* discharge coefficient

*M* mass flow rate, kg/s

*P* pressure, kPa

*Pb* back pressure, kPa

*R* mass flow rate ratio

*T* temperature, K

*v* specific volume, m3

*α* homogeneous void fraction

*η* pressure ratio

*ρ* density, kg/m3

*ω* compressible flow parameter

ID inner diameter, m

MAP maximum allowable pressure, kPa

MAWP maximum allowable working pressure, kPa

OD outer diameter, m

PCB printed circuit board

LIST OF APPENDICES

**APPENDIX TITLE PAGE**

**No table of figures entries found.**

# Chapter 1

## INTRODUCTION

### Background

Eye Tracking is a collection of research engineers that provide access to eye movement information, its location within a certain time range and (if any) view. For example, you can use the information obtained in this way. Perform research on the use of interfaces, read texts or the effectiveness of a commercial announcement. Every day our vision is used intensively for many different purposes: reading and watching entertainment companies and implementing and learning new things. But now we are aware of the complexity of the system. Visible people or eye movements are a technique that has been used for over 100 years in areas like psychology, medicine, interactions between people and computers, marketing and many more.

### Problem statement

The purpose is to develop and apply a system that tracks the user's computer eyes in real time Bales the system aims to estimate the current focus of the person sitting in front of the screen. The most important assumption of the author's work is comprehensive development, which is a required application that does not require any special machinery to operate. The optimal solution must be made up of a personal computer and a single web camera. There are many trading systems in the market that allow you to pursue your vision. One of the disadvantages of the existing solutions is the need for special equipment, from dedicated head-mounted devices for pan-Arab cameras, and it is easy to see the advantages of a system that does not require specialized equipment, Another important issue is that the system can operate in real time, allowing control of focus points, which requires selection algorithms with a very large computational complex. The image is converted to variable lighting conditions M can meet the specific requirements. This is evident from the fact that there is no commercial solution to this problem.

# Chapter 2

## LITERATURE REVIEW

### Existing Solution

There are many ways to record human visual activity, from regular direct observation through gaseous mechanical methods, ending with power exploration teams between the sides of the eyeball. The research has been conducted, which means that there are many very different solutions to each other. Choosing the best method depends on the purpose of the searches. The purpose of a system is to determine the accuracy, accuracy, frequency measurements, ease of use and prices required. Eye tracking systems can be divided into several groups: the location of the device for the head (mobile and non-mobile), the type of data obtained, and the way to determine the KS point (or eye movement itself). The following is a general description of the methods used to measure the position of the eyeball.

### Type of data obtained

Eye tracking researchers are interested only in eye movement information, but not in reference to the point of descent of space in space (the point of the priesthood is a place

In the field of vision of the person examined, which is considered at a given time). For this type, the most important information is about the position of the eye for the starting point. These data have absolute character values. This technique is also known as eye tracking (orbital eye tracking). An example of the above-mentioned device is the Polish invention JAZZ-novo fig.1, which was created in cooperation with the Institute of Bio-Control and Biomedical Engineering at the Polish Academy of Sciences in Poland.

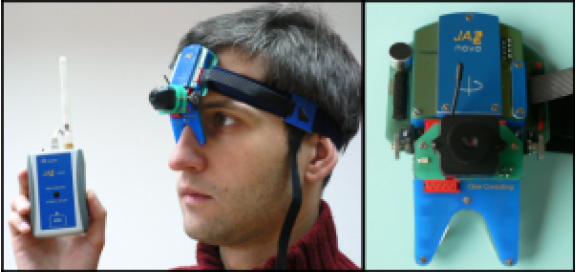


Figure 1JAZZ-novo measuring system

The information collected by this method is often used to investigate the vestibular device, the eye examination and the diagnosis of dyslexia. There are also practical applications in military aviation, which provide information on the interaction of the pilot with the cockpit of the aircraft.

.

### An important reference point

### In a much wider range of applications, there are systems that return data about the relative location of the eye, ie, in relation to the point of detection of books on the screen in space. The test results are usually displayed in the image of heat maps (Figure 2.2). This is possible thanks to the use of advanced algorithms that monitor eye position and combine the father's point of view with the visual phase. All environments assigned to this task (gaze tracking systems) must be calibrated in an advanced manner. The global leader who offers such a solution is Tobias.

### Three-dimensional direction of sight

Three-dimensional orientation for viewing.

The 3rd category of eye track is known as oculogyric and 3Dimensional. In accumulation to the horizontal statistics and the perpendicular divergence of eye, the information about rotation (turning round) about its axis. For few machines, we can manage the movements and location of head, with the help of that we can get the data of location of eyeballs( rotational axis of eyeball). The products of this problem are the GCSMI “German company SensoMotoric Instruments” GmbH. It's a bold 3D device that follows your eyes.

****

Figure 2Eye test results represented by the heat map



Figure 33D Video system

### Methods for measuring the position of the eye

#### Electro-Eye tracking

The diversity technique involves measuring the skin around the eyes attached to the potential use poles. It can be a place for the eye to detect a pinpoint, when they remembered thanks to the skin, a small amount of variations in the potential accuracy. But this is not enough to require the use of the skin, using the art of Figure 2.4 with the unpleasant or exists closest to the electrodes to be seen. For this is the manner of the most widely used solution to the 40 years, it is not yet you will find on the application. A very different way of measuring the tension of the muscles of the eyes of the competition. Depending on the breadth of the signal that traverses the distance between the eyes rapid changes in the site (also called "saccades"). This is mainly thanks to the potential on the front and the eyeball is different from the back of your potential. System's sensible sound is mainly based on an act other muscles his face. electric eye tracking | there is often used in medicine. the changes to be made in the fields of electric charge of eye movement control means allows in its place.

#### Technician using contact lenses

The use of special contact lenses made it possible tow accurately evaluate the eyeball movement.  
there are lenses such as small induction coils. Making the lens positioning precise it is done by recording the changes E-M field caused by eye movement. Problem arises in head movements. Major disadvantage is restriction of traffic and non-bulky devices, which makes it to be only used for laboratory purposes.

#### Optical techniques based on video recording

##### Using infrared light

Infrared illumination of the eye facilitates some local dog walking in the optical axis [4]. RENICO almost all This phenomenon reflects the light that makes the room for each tablet on a white figure 4 similar to the red eye. Infrared light source of infrared light outside the optic axis of the eye, Figure 4, the method consists of the dark place, eye cameras identifying the center of the pupil and the location between the bright source of the cornea from the infrared light. Reecisents of light reflected various structures called receptor imaging (Fig. 5). Marks light reflection on the eye of the developer with the corneal vision (it was called the 1 Light görüntü image receptor P1) to mention that the pupil's Middle is high. The accuracy of the measurement can be increased price and the number of registered points of the users in the quartile, is the device (gleam). There are also devices, and the correct measurements have two receptors (double eye of the developer's eye) for use in light images of the developer's corneal surface

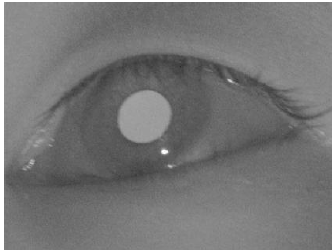


Figure 4 (a) Bright source effect caused by the source

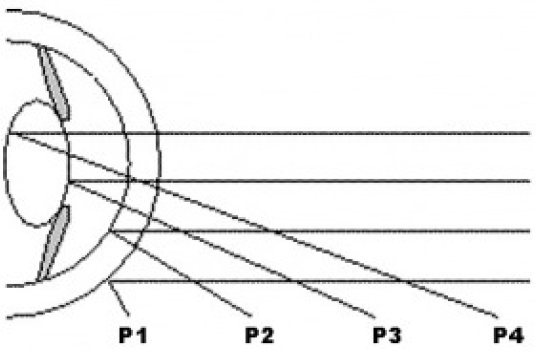


Figure 5Four types of Purkinje images

.i,e the first image receptor P1 on the rear face of the lens (the fourth image receptor P4) is based. The special effects or the beard generally support teething (bit-bar) thanks to the use of tuition.

#### Without using infrared light

The image is recorded using a standard camera. The reason for this activity is that the position and shape of the cornea are measured according to the standard part of the group (eg, the angle of the eye). Corneal mucus is usually a damaged element. The solution, which is part of the following verse, places the apple on the ground with a horned envelope. However, the contrast between the cornea is much smaller than the rest of the mouth and eyes. Otherwise, it uses a neural network. Know with foot max. This kind of system calibration procedure takes a long time. You need to judge a large amount of images to learn the correct neural network. In fact, most languages used by the camera assume that the user has the actual test header. To ensure free movement of the head, the current position of the head (displacement and short circles) is more accurately determined by the developer.

Consider the eye and the point on the screen for a certain moment of the person. The calibration of these systems takes a long time. It requires a large number of input images for proper learning of the neural network. Systems based on the use of a standard video camera generally assume that the user still maintains an ideal head during the test. To allow freedom of movement of the head, it is necessary to determine the precision in the implementation of the current methods of head position (compensations and short circuits).

### 2.3. Review of existing solutions

There are a full range of eye tracking systems in the market. The leading products are Tobii and SMI. These are remote systems, that is, no hardware is installed on the user's head. Its activities rely on the use of camera images in the infrared range and infrared light source. The current view is the difference between light and light. However, there is no trading system that runs on a standard application camera. COGAIN, which works among other things to create a low-cost eye tracking system using a standard camera. Currently, the only available procedure is to handle the sight examination using a standard open gazer camera. It is an open source program. Its main drawback is assumption, the user header is completely silent while running the program. Even small movement makes it necessary to reset the system.

### Theoretical basics

#### Visual system

#### In order to engineer an eye tracking ystem it is foremost to understand the core issues regarding the human’s visual system. It will help in understanding the methods of designating current direction if sight moreover to understand and identify existing constraints.

#### Description of the visual system

The human eye is made up of a lens with variable focal length and adjustable, a complex structure that regulates the diameter of the aperture through which light and light enter the deep eye of the retina. The photoreceptors in the retina change the light to electrical potential, leading to nerve impulses that are transmitted to the cortex by the optic nerve. The retina consists of two types of receptors: suppositories and prisms. The brushes are very sensitive to light intensity, allowing to watch in environmental conditions with very low light intensity. However, these receptors can not record colors, making everything seen in a gray night. Unlike suppositories, they need more light to produce nerve impulses, but they are able to recognize colors. Brushes contain three types of filters sensitive to different lengths of light: red, green and blue. Due to the large variation between the hard core and the manual, it is possible to evaluate the direction of the display. One of the reasons for this construction of the eyes is that "Human is a social being, and the direction of sight is very useful during communication.Location »Yellow is responsible for the sharpness of the image. It is the largest conical diameter of about two meters in the center of the retina. On the other side of the network, men go from 15% to 50% of the light receptors. This makes the visualization of the image visible to a degree. It is not possible to focus your eyes on an object smaller than a yellow "dot," which makes it impossible to determine the orientation of the vision more accurately than one's "speed".

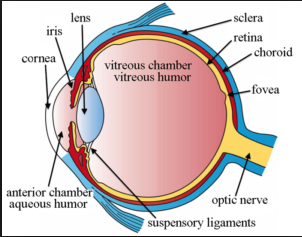


Figure 6 Construction of the eye

#### Eye movement

The two basic purposes of eye movements are to stabilize the image on retina to compensate for head movements or any movement of object in the sight of field, setting relative environments on retina.

#### During normal human activities, eyes remain in the state of theocracy, that is, relaxation. In the book, visual information is extracted from the environment. The length of the book depends on how you handle the information. The result is about 0.15 seconds to 1.5 seconds on average, where the sacca operates from 4 to 6 times per second, ie a very fast movement and jump to change the location between the next book. Sakada generally lasts from 0.03 seconds to 0.06 seconds.

#### Optical flow

Optical flow is a vector field that allows you to transform a given the image in the sequence in the next image of this sequence, by moving the areas from the first the image (for which this field was specified), according to the corresponding vectors of this field on the second picture. In short, optical flow is a set of translations (in the form of a field),which transforms the given image in the sequence into the next image in the sequence [3]. There are many methods for determining the optical flow. They can be divided into three main ones group:

Gradient methods based on the analysis of derivatives (spatial and temporal) intensity Image, Frequency-domain methods based on the detection of imaging information in the field ± ones Correlation methods based on the appropriateness of image areas. Algorithm of Lucas-Kanade The algorithm Lucas-Kalman was used in the work. This is a method gradient.

The operation of the algorithm is based on three basic assumptions:

1. The image brightness does not change much between successive sequence frames.

2. The speed of motion of objects in the image is small.

3. Points that are slightly close to each other move similarly.

The brightness of the image is determined by the time-dependent function.

***f*(*x; y; t*)*\_I*(*x*(*t*)*; y*(*t*)*; t*)**

Requiring that the image brightness does not change significantly over time is shown by the equation:

***I*(*x*(*t*)*; y*(*y*)*; t*) = *I*(*x*(*t* + *dt*)*; y*(*t* + *dt*)*; t* + *dt*)**

This means that the intensity of the traced pixel does not change over time:



Using this assumption, you can write an optical flow condition meaning thatthe vector of velocity in the x direction, and v by the vector of velocity in the direction y



Unfortunately, this is an equation with two unknowns. To solve them, you need additional onesinformation. The third assumption at this point is used: points locatedthem move in a small distance similar to each other. Thanks to this solution of the equation forone point uses additional pixels surrounding it. To solve the system equals «using the least squares method.The assumptions of this method make it able to detect only very low muscle traffic.Between subsequent frames. To improve its operation, a pyramid of images is used. It is based onperforming the algorithm successively in the image with reduced resolution.

# Chapter 3

## DESIGN AND METHODOLOGY

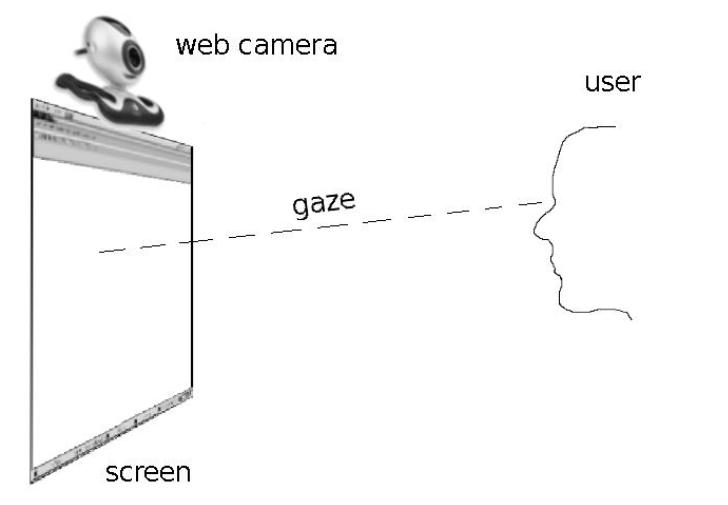


Figure 7: System diagram

It offers a vision tracking system based on the recording of eye images with a standard web camera. The diagram of the system appears in Figure 7. For simplicity, the camera is placed on the screen at its center. Most commercial systems require that the camera be placed at the bottom of the screen, which gives you a slightly better eye image. However, the purpose of this work was to find a global solution. The webcam is usually above the screen, and an example could be laptops with a built-in camera, where you can not change its location. The purpose of the action is to determine the point on the screen that the user is currently looking for. To obtain this data, it is necessary to determine the relative location of the eye and head. The first stage is to initialize the algorithm, during which the header model is created. After initialization, trace the trace phase. To determine the position of the 3D head, the optical flow between successive frames obtained from the camera and the POSIT algorithm is used [13]. A change in the direction of sight is determined by studying the differences in the distance between the center of the eye and the center of the eye. This data is obtained using a complete set of methods to process the images described in the rest of the work. For the results presented, a complete set of algorithms was tested, part of which was presented in the following work.

### Tracking Algorithms

In this section the algorithms will be discussed that have been used for head and pupil detection to understand their working.

#### Tracking head position

A large number of available vision tracking systems require limited head movement. Is not it an appropriate or practical solution? A man is unable to keep his head still in full condition for a long period of time without using the foundation on which to stand the beard or end. These devices are used in ophthalmology during eye examination. The involuntary movement of the head has a significant impact on breathing. It will be possible to determine the direction in which the person sitting on the computer without the restriction of the main motion is restricted, it is necessary for a very accurate 3D face tracking position. Determining the position of the head is a very important area of ​​research on human-computer interaction (HCI). There are many ways to estimate placements with a single camera. These methods can be divided into two main groups: a model of the head and face features. Ways to use the model to determine the position and estimate 2-3D reliability between features. The situation is determined by using these relationships. Face-based methods form the relationship between the 3D position and certain characteristics of the facial image. Identify these dependencies using a lot of images from those who train with a well-known front position to train the neural network. The disadvantage of this method is the unusual work of new faces that were not used during image training. Two different approaches will be presented. The first of the works [10] uses the AAM algorithm [28]. This is one of the most popular solutions to track key positions in recent years. The second method is the method using the head sinusoidal model and assumes an algorithm [13] to determine the current position. At work [30] a combination of the two methods was presented.

#### Active Appearance Model

The Active Appearance Model (AAM) [28] is a way to detect and track any object whose models are created during the learning phase based on a set of correctly prepared input images. AAM has a wide range of applications, from medical images to segmentation, facial recognition and head tracking. For help, the algorithm can be designed on any object known for its shape and texture. AAM is a method of data control, which means that it is not necessary to specify parameters that work manually. The algorithm automatically adjusts the data during the initialization process. Its main drawback is the fact that the success of the process depends largely on the appropriate selection of a set of images with the body shape used during the form creation phase. Creating teaching data can consist of hundreds of images, and the right shape of the body can be a daunting task. Of course, it is possible to automate the process of modeling in a learning dataset, but this problem goes beyond the scope of this work. For model construction, the Basic Component Method (PCA) is used, and is assumed to be required after the model parameters have a gusowski distribution, which in extreme cases may result in a way that the model will be subject to unrealistic deformity. The active appearance model consists of a fixed texture and a shape model.

##### Static shape model

An effective model is a structure that contains information about the medium as a particular object type (for example, facial) and data that describe more fashions and features thus observed for general instruction. Form form can be modified by algorithms that attempt to adapt to the actual shape, while not allowing natural deformation. Create a form that begins to build on the master of objects, that is, a set of labeled points represents a certain shape. This is called point distribution model (PDM point distribution model). Minutić should be placed in appropriate places in all N images learning. In this way we get a set of teaching methods, stored in the form of vector containing the coordinates of the characteristic point. Data relating to the size and location of objects are removed from the vector by standardizing special procedures, so that they are only shape information. The automatic positioning of the highlights in the images is a very complex problem, so the safest way to get a sample set of forms is to put a guide, which will certainly be difficult and time-consuming. A shape is defined as a set of 2D points that creates a cross-linking grid on the object being scanned. Reference points (reference points) can be placed in the image automatically or manually by "user." Mathematical record forms are expressed by two-dimensional vectors.

#### The algorithm of face and eye detection\

1.Search the image using the EMgu class to discover the face. The result is the location.

Find all faces Larger face is included for additional analysis size, that is, the person closest to the camera.

2. Using face locations previously selected, the following EMgu class is used to expose the eye. To reduce CPU load, only the upper half of the face is searched. It also reduces the risk of improper eye detection, p. For example, place your mouth or nose.

3. Check the size and mutual position of the face and eyes. The size of both eyes and their horizontal position should be close to each other. If they are not met, the entire detection requirements are implemented from the beginning in the new cage.

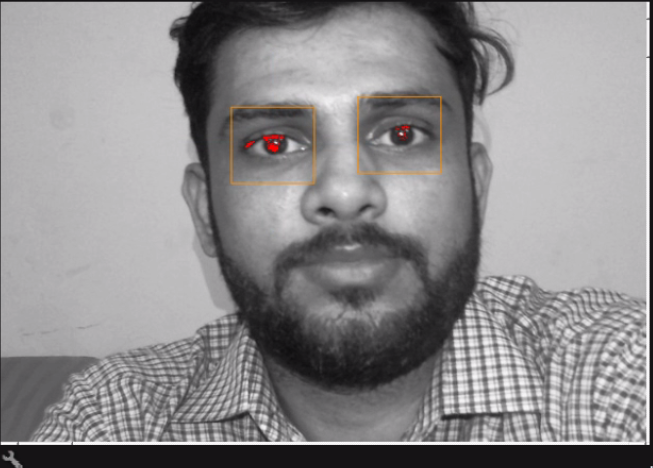


Figure 8Face and eye detection

The result of the algorithm is shown in Fig. 8. Combining EMgu class operation for face and eye detection results in much better results. Based on the experience, "it can be concluded that the use of HAAR only for facial detection can get erroneous results, for example, the region of the image where there is no face may be incorrectly classified. This situation usually prevails in the case of lighting conditions other than those on the images used to train the EMgu class. Especially the wrong result was achieved with the side lighting of the face. Combining the operation of both the ducks layer and checking the mutual location of the eyes and the size of their eyes gives you a more reliable process of algorithm. The price you have to pay for the fact that sometimes there is a face in the input image will not be found. However, this is not a big problem because the program uses video sequences and you can ignore some initial frames. The facial detection algorithm is repeated frequently until the face and eyes are detected and their mutual position conditions are met.

#### Haar Feature-based Cascade Classification

Recognizing objects using cascaded classifiers was proposed by Viola and Jones and Lienhart, and Maydt improved it slightly. To train many positive and negative image classifiers, the machine learning approach is implemented. The classifier is then used to recognize objects in other images. In order for the classifier to find objects with dimensions smaller than the original image, each position of the image must be examined using a classifier. The advantage of the classifier is that it can be scaled if you need to find the object instead of scaling the original image. If you do not know the size of the object you have to find, you need to scan several times at different scales. Cascaded classifiers consist of many simpler classifiers and are applied following the target area of ​​interest until one of them is rejected or all has passed. The Haar like features are shown in Figure 9. The characteristics used by the classifier depend on shape, position, and scale.

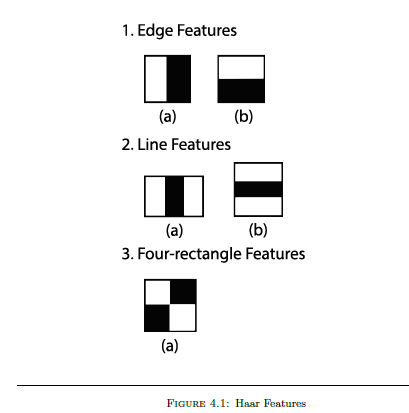


Figure 9 Haar features

#### Template Matching Algorithm

In computer vision, the term coincidence with the template is a technique to find image areas similar to the image of the template. In this technique, there are two main components:

• Source image

• Template image

The source image (I) is the original image in which the template is applied, to find a specific region

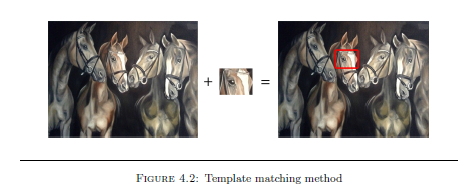


Figure 10 template matching method

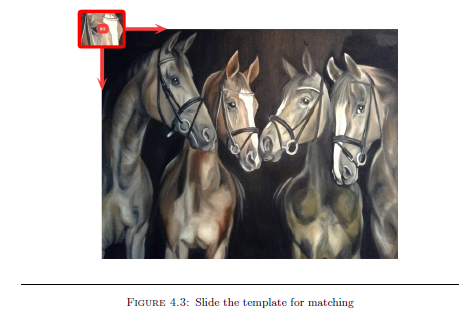


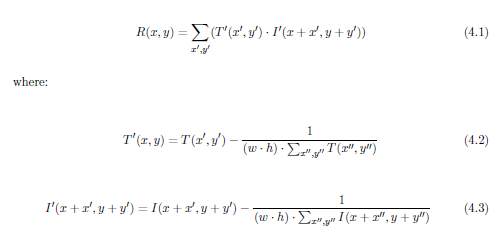
Figure 11 slide the template for matching

. The image of the template (T) is the image used to compare with the source image to detect the best matching area as shown in figure [4.2]. To select the matching area, the image of the template must be moved over the source image, one pixel at a time, as shown in Figure 10. In each pixel, several

The calculations are made to show if the point of the template image matches the point of the source image at that specified time. To do this, the value of the scale is stored for comparison in the matrix (R).

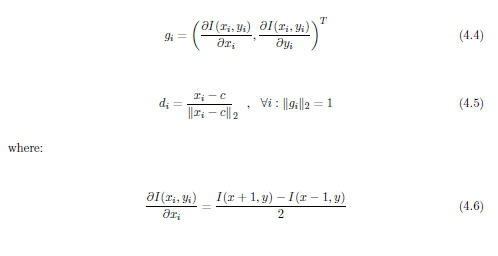
There are several ways to match templates in OpenCV. One used for this thesis is the correlation coefficient [22]. I (x, y) is the value of the source image pixels and T (x, y) is the value of the template pixels in the location (x, y). The width of the template image is displayed in pixels with the height of the template image at the value of h. T is the average value of the pixels in the template and I mean the pixel value in the current window of the source image, where x = 0 ... W - 1 and y = 0 ... h

The result matrix R is calculated by adding the product T '(x', y ') and I' (x + x ', y + y') to each overlap position shown in Equation 4.1.



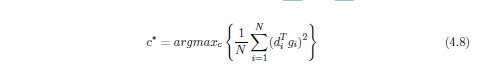
#### Timm’s and Barth’s Algorithm Using Image Gradients

The method of determining the center of the pupil is Timm Bath [12]. To find the center of the circular object, in this case the pupil, the image gradient vector is used. Possible centers depend on the orientation of these vectors. If c is a possible center and gi (Equation 4.4) is the gradient vector at location xi, di (Equation 4.5), ie the normalized vector must be in the same orientation as the gradient vector gi.





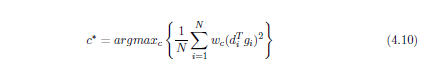
By finding the dot product between the normalized vector di and the gradient vector gi in the pixel xi, {1,. ,, N} (Expression 4.8).



To get the same weight for each pixel location, both the di vector and the corresponding gi must be scaled in length to improve the illumination vector and contrast. For eyeglasses and eyelids, the possible central evaluation may be incorrect. To correct this, the weight “wc” is applied, and instead of a bright one, a center with a possibility of darkness is selected. “wc” is the gray level of the inverted input image I\* as shown in (Expression 4.9).



In addition, Gaussian filters are used to avoid reflections from glasses and generally bright locations. Possible centers are calculated by weights as shown in (Equation 4.10).



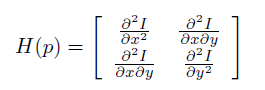
This is the whole information required when the image gets the image of the eye. Sometimes hair, eyeglasses and eyebrows show different images in comparison to the pupil. Therefore, post-processing steps are helpful. A threshold is applied to remove all gradient values from the image boundary. The maximum of these values is used to determine the center of the pupil.

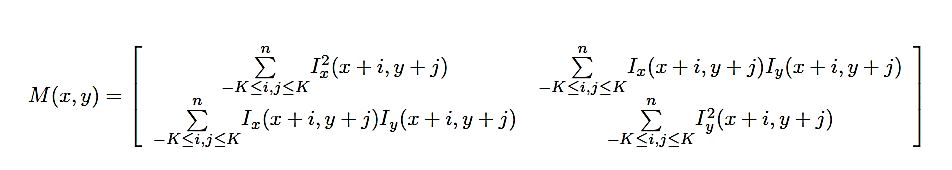
#### Determination of tracking characteristics

Commercial systems for the detection of facial expressions and movements of the head using markers made of reflective infrared light material adhered to the face. This solution is very complete, however, it requires expensive equipment. The action assumption was to create a system without necessity.

Convergence of complex devices. The solution provided only depends on a standard web camera without using additional debug tags to run the algorithms.

Luke's intermittent optical flow was used to track the change in head position described by Kalman in the theoretical part. To take action, you must establish a set of features that will be tracked and repositioned in the following boxes for the video sequence. The characteristics must be selected with each other so that it can be determined that they change their position in cascaded frames in an unequivocal manner.



The M Hesjanic autocorrelation matrix is ​​determined by summing the second serene valuesin the vicinity of a given point

The Lucas-Kalman algorithm works well if the plotted points are present where the sharp edges are present. Choosing the right feature set is a very important issue. He has had a significant impact on the other path of change in the position of the face. The literature can find many different ways to identify good tracking points. One of the basic criteria for selecting the appropriate method was arithmetic complexity. From this assumption, the program must work in real time, so you can not afford a complicated choice that would increase the load on the processor. Harris [12] gives one of the most commonly used tip motifs. This de-thread is based on a matrix of second-degree derivatives (HSS) of the density of the image. Burlap is defined in p (x; y) as follows:

The edges are located in places where the Hessian self-correlation matrix has two values. This means that the fabric around a particular point changes dramatically in two independent directions. By using only their own values, the specified edges do not change when the image rotates. Find local limits for self-association of the input image. You can easily obtain points that can be tracked using optical flow. It is very important to distribute certain characteristics uniformly on the surface: any similar object. This can be achieved by limiting the minimum distance between contiguous points. Such a supposition is necessary because in the event that the tissue of the Hessian autocorrelation object is a great maximum in one place, the result will be the location of most of the features around this limit. In this case, an algorithm that determines the position of the 3D object can produce incorrect results. The method shown is based on the identification of natural facial features, which are easy to trace with a flash. The algorithm looks for local maxima, making them adaptable, depending on the appearance of the person's face, and looking for optimal tracking points. Such a solution is more universal and offers better results than the specific "characteristics" of the face, such as the corners of the mouth, the corners of the eyes, etc. The configuration of fixed characteristics is not always possible. In the case that we assume that we are using, for example, the angles of the eyes, the algorithm that looks for these characteristics, does not wait for the fact that a person has been wearing glasses, it can worsen the result and even the incorrect results.

#### Algorithm for determining traits for tracking

1. Select the search area. The appearance of features is limited to the face of the selected area during the previous phase. The area that is not taken into consideration is not taken into account because the eyes were present because the eyeball movements could interfere with the correct position of the head of appointment.

2. Define characteristics. The Harris algorithm was used to determine properties [12].

3. Jump on points very close to each other. To increase efficiency eliminated.

There are features located very close to each other.

#### Use of reference frames

Using the previous method, information about changing the position of selected faces is displayed only between the next sequence of video segments. This method gives satisfactory results in cases where the number of tire treatments is important. When a long sequence occurs, the accumulation of errors becomes important in judging the position of the entity between successive frames. The solution to this problem is the application of the method provided in [19]. Approach "is the use of a series of reference images, which is a reference image selected from the frame with the current head and all facial features of the face, and when the position is at the top of the window eliminates the cumulative errors of positional address errors. so that the behavioral head of the tracking algorithm does not get worse Over time, it determines when the information is only used to change the entity's position between successive frames during system initialization, create a reference and startup image, and status automatically add a new image when the status reference is monitored with the current record's current position record, and create a new image at the location specified in the algorithm The algorithm works automatically If you use the operating algorithm framework.

## Implementation

Vision-based eye blink systems have a lotof possible applications, such as fatigue control, Computer interface and lie detection. It does not matter what the goal of the system is to develop an improved algorithm reliable, stable and works in real time with changing lighting terms.

The vision-based blink detection system is based on the components available for sale: anA PC or laptop is of average quality, Webcam. Lower face images (320 \* 240

Pixels) processed at approximately 30Fps. The eye blink detection algorithm consists of four steps the main steps includes (1) face detection, (2) eye area

Extraction, (3) eye detection and (4) eye blink classification. Face detection is performed using features similar to Haarr and a set of improved tree workbooks.

The location of the eyes depends on certain geometric relationships known to the human face. Eye blink detection is done using template comparison technique. All

The steps of the algorithm are described in greater detail in

Section (4.1 - 4.4). The algorithm makes it possible to detect eye blinking,

Evaluation of the duration of the blink of an eye and on this basis the classification of eyelash in kind as spontaneous or voluntary

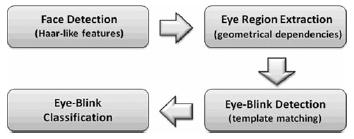


Fig. 4.1 Scheme of the proposed algorithm for eye-blink detection

## 4.1 Face detection

Face detection is a very important part of development of Wink detection algorithm. Due to the fact that the face detection site is arithmetically expensive, therefore consumes a lot of time, this procedure is only performed through System configuration and in cases where it is continued. The face is "lost". Therefore, for a system that works in real time, the selected method should work quickly (less than

30 for each single image at 30 frames per second). At the end

On the other hand, the accuracy of the specific focus is also at the important compensation must be between the high detection rate, fault detection and error rate. The chosen the method must work strongly under different lighting conditions.

Different facial expressions, head formation, partial, Face blockage, having glasses, facial hair and. Different hair styles have been suggested by many solutions. Facial Detector. Can be divided into: (1) Knowledge-based methods use simple rules for describe properties of facial symmetry and geometry, relationships between facial features [28], (2) feature methods depend on the detection of mouth, eyes,

Nose or skin colour [20, 21], (3) methods of combining the mold based on the calculation of the relationship between inputs the image and patterns of faces stored [22] and (4) based on the model. Methods in which algorithms are trained in models.

Using neural networks [23], supporting vector machines(SVM) [24] or hidden Markov models (HMM) [25]. InanAlgorithm developed, the method derived from the template. A matching set developed by Viola and Jones [6], has been modified

By Leinchart and Maydt, and carried out According to which it was used.

Similar properties are calculated with Haar by dispersing

Image with templates (Figure 4) of different size and direction. These prototypes can be grouped into 3 categories: border, line and centre around the masks. Everyone

The mold consists of two or three black and white rectangles.

## 4.2 Eye-region extraction

The next step of the algorithm is to locate the eye area picture of the eyes in the face.

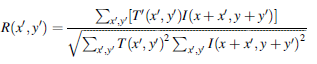
It is based on some engineering dependencies known to the human face. Traditional rules of proportion. The face appears as divided into six equal squares, two by for three [7]. According to these rules, eyes fall. About 0.4 way from the top of the head to the eyes, The image of the extracted ocular area is further away. Pre-treatment of eye blink detection performance. The local ocular area is extracted from the facial image. It is used as a model for further eye tracking through matching templates. The extraction of the eye area is performed only in system configuration and in

Cases where face detection is repeated.

### 4.3 Eye-region extraction

The detected eyes are tracked using a normal reciprocal relation. Method (1). User Image Template. The eyes are automatically obtained during initialization of

The system



Where R correlation coefficient, T image template, the original me

The coordinates of the image are x and pixel. Coefficient of correlation.

It is a measure of the current eye resemblance.

## 4.4 Eye-blink classification

The change in correlation coefficient is analysed in time to detect voluntary winks greater than 250 milliseconds. If the parameter value is lower of the predefined TL threshold value for two consecutive processes in the pictures, the eye blink was detected. The blink of the eye is found if the correlation value of

Parameter is greater than TH threshold value. Threshold values were TL and TH determine empirically. If the duration of the detection longer than 250 ms and shorter than 2 seconds this flicker is considered a voluntary "control".

### 4.5 Code

using Emgu.CV;

usingEmgu.CV.Structure;

usingEmgu.CV.Util;

usingEmgu.CV.CvEnum;

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Windows.Forms;

using System.Drawing.Drawing2D;

usingSystem.Collections;

using System.IO;

usingSystem.Drawing.Imaging;

usingSystem.Globalization;

usingSystem.Speech.Synthesis;

namespaceCameraCapture

{

publicpartialclassFaceDetection : Form

{

SpeechSynthesizer synthesizer = newSpeechSynthesizer();

int temp = 1000;

int op = 1;

boolrightwink = false;

boolleftwink = false;

int wink = 0;

stringfl;

int d1, d2;

intrwink = 0;

intlwink = 0;

intarg = 1000;

privateCapturecapturedCaptureFromWebcam; privateboolisCaptureInProgress;

privateHaarCascadefaceHaar;

privateHaarCascadeeyeHaar;

privateHaarCascademouthHaar;

privateboolcbFace;

privateBitmaporiginalImageFromPics;

privateBitmaptemporaryImageCopiedFromOriginal;

privateBitmapfinalImage;

privateboolimageFit = false;

MCvFontfaceFont = newMCvFont(Emgu.CV.CvEnum.FONT.CV\_FONT\_HERSHEY\_SIMPLEX, 3.0, 3.0);

MCvFonteyeFont = newMCvFont(Emgu.CV.CvEnum.FONT.CV\_FONT\_HERSHEY\_SIMPLEX, 2.0, 2.0);

MCvFontmouthFont = newMCvFont(Emgu.CV.CvEnum.FONT.CV\_FONT\_HERSHEY\_SIMPLEX, 2.0, 2.0);

publicFaceDetection()

{

InitializeComponent();

}

privatevoidopenButton\_Click(object sender, EventArgs e)

{

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

originalImageFromPics = newBitmap(openFileDialog.FileName);

originalImageFromPics = originalImageFromPics.Clone(newRectangle(0, 0, originalImageFromPics.Width, originalImageFromPics.Height), PixelFormat.Format32bppArgb);

temporaryImageCopiedFromOriginal = newBitmap(originalImageFromPics);

intnewWidth = Math.Max(this.PreferredSize.Width, 305);

intnewHeight = this.PreferredSize.Height;

newWidth = Math.Min(newWidth, Screen.PrimaryScreen.Bounds.Width \* 2 / 3);

newHeight = Math.Min(newHeight, Screen.PrimaryScreen.Bounds.Height \* 2 / 3);

this.Size = newSize(newWidth, newHeight);

imageFit = false;

}

catch

{

MessageBox.Show("Cannot open the file.");

}

}

}

publicclassAutoClosingMessageBox

{

System.Threading.Timer \_timeoutTimer;

string \_caption;

AutoClosingMessageBox(string text, string caption, int timeout)

{

\_caption = caption;

\_timeoutTimer = newSystem.Threading.Timer(OnTimerElapsed,

null, timeout, System.Threading.Timeout.Infinite);

using (\_timeoutTimer)

MessageBox.Show(text, caption);

}

publicstaticvoid Show(string text, string caption, int timeout)

{

newAutoClosingMessageBox(text, caption, timeout);

}

voidOnTimerElapsed(object state)

{

IntPtrmbWnd = FindWindow("#32770", \_caption);

if (mbWnd != IntPtr.Zero)

SendMessage(mbWnd, WM\_CLOSE, IntPtr.Zero, IntPtr.Zero);

\_timeoutTimer.Dispose();

}

constint WM\_CLOSE = 0x0010;

[System.Runtime.InteropServices.DllImport("user32.dll", SetLastError = true)]

staticexternIntPtrFindWindow(stringlpClassName, stringlpWindowName);

[System.Runtime.InteropServices.DllImport("user32.dll", CharSet = System.Runtime.InteropServices.CharSet.Auto)]

staticexternIntPtrSendMessage(IntPtrhWnd, UInt32Msg, IntPtrwParam, IntPtrlParam);

}

privatevoidfitButton\_Click(object sender, EventArgs e)

{

privatevoidDetectOrgansInFace(object sender, EventArgsarg)

{

#region

Image<Bgr, Byte>imageFrameCapturedFromCamera = capturedCaptureFromWebcam.QueryFrame();

if (imageFrameCapturedFromCamera != null)

{

Image<Gray, byte>grayOfImageFrameCapturedFromCamera = imageFrameCapturedFromCamera.Convert<Gray, byte>();

if (cbDetectFaceFile.Checked).

{

var faces = grayOfImageFrameCapturedFromCamera.DetectHaarCascade(faceHaar,

1.2,

5,

HAAR\_DETECTION\_TYPE.DO\_CANNY\_PRUNING,

newSize(25, 25))[0];

if (faces.Count() == 0) { panel1.BackColor = Color.Red; }

foreach (var face in faces)

{

imageFrameCapturedFromCamera.Draw(face.rect, newBgr(Color.Blue), 10);

intfaceLableX = face.rect.X;

intfaceLableY = face.rect.Y;

Point p = newPoint(faceLableX, faceLableY);

String s = "Face";

imageFrameCapturedFromCamera.Draw(s, reffaceFont, p, newBgr(Color.Blue));

.

{

panel1.BackColor = Color.LightGreen;

var eyes

= grayOfImageFrameCapturedFromCamera.DetectHaarCascade(eyeHaar,

4,

3,

HAAR\_DETECTION\_TYPE.DO\_CANNY\_PRUNING,

newSize(2, 1))[0];

foreach (var eye in eyes)

{

imageFrameCapturedFromCamera.Draw(eye.rect, newBgr(Color.Red), 3);

inteyeLableX = eye.rect.X;

inteyeLableY = eye.rect.Y;

Point p2 = nzwPoint(eyeLableX, eyeLableY);

String e = "Eye";

imageFrameCapturedFromCamera.Draw(e, refeyeFont, p2, newBgr(Color.Red));

d1 = eye.rect.X - face.rect.X;

d2 = eye.rect.Y - face.rect.Y;

if (eyes.Count() == 1)

{

if (d1 >= 35 & d1 <= 70) { rwink++; }

if (d1 >= 100 & d1 <= 230) { lwink++; }

}

else { rwink = 0; lwink = 0; }

richTextBox1.AppendText("=======================================================================================\n");

richTextBox1.AppendText("distance x=" + d1 + ", y=" + d2 + "\n");

richTextBox1.AppendText("face rect \n");

richTextBox1.AppendText("face rect x = " + face.rect.X + "\n");

richTextBox1.AppendText("face rect y = " + face.rect.Y + "\n");

richTextBox1.AppendText("eye rect \n");

richTextBox1.AppendText("eye rect x = " + eye.rect.X + "\n");

richTextBox1.AppendText("eye rect y = " + eye.rect.Y + "\n");

richTextBox1.AppendText("right wink :" + rwink + " \n");

richTextBox1.AppendText("left wink :" + lwink + " \n");

richTextBox1.ScrollToCaret();

}

if (eyes.Count() == 0)

{

panel1.BackColor = Color.OrangeRed;

label1.Text = "blink time im ms:" + wink.ToString();

wink += (1);

}

label2.Text = op.ToString();

if (eyes.Count() >= 1 &eyes.Count() <= 2)

{

panel1.BackColor = Color.Green;

if (rwink>= 2 &&lwink == 0) { if (op >= 2 & op <= 6) { op--; rwink = 0; lwink = 0; } }

if (lwink>= 2 &&rwink == 0) { if (op >= 1 & op <= 5) { op++; rwink = 0; lwink = 0; } }

if (wink >= 3) {

if (op == 1) { synthesizer.Speak("I NEED DOCTOR ");

wink = 0; AutoClosingMessageBox.Show("I NEED DOCTOR ", "BLINK DETECT", 3000);

}

if (op == 2)

{

synthesizer.Speak("I NEED FOOD ");

wink = 0; AutoClosingMessageBox.Show("I NEED FOOD ", "BLINK DETECT", 3000);

}

if (op == 3)

{

synthesizer.Speak("I AM READING NEWS ");

wink = 0; AutoClosingMessageBox.Show("I AM READING NEWS ", "BLINK DETECT", 3000);

Application.Idle -= DetectOrgansInFace;

Form1 obj1 = newForm1();

obj1.Show();

}

if (op == 4)

{

synthesizer.Speak("I WANT TO GO TOILET");

wink = 0; AutoClosingMessageBox.Show("I WANT TO GO TOILET ", "BLINK DETECT", 3000);

}

if (op == 5)

{

synthesizer.Speak("I NEED WATER ");

wink = 0; AutoClosingMessageBox.Show("I NEED WATER ", "BLINK DETECT", 3000);

}

if (op == 6)

{

synthesizer.Speak("I NEED NURSE ");

wink = 0; AutoClosingMessageBox.Show("I NEED NURSE ", "BLINK DETECT", 3000);

}

}

if (op == 1)

{

button1.BackColor = Color.Green;

button2.BackColor = Color.Red;

button3.BackColor = Color.Red;

button4.BackColor = Color.Red;

button5.BackColor = Color.Red;

button6.BackColor = Color.Red;

}

if (op == 2)

{

button1.BackColor = Color.Red;

button2.BackColor = Color.Green;

button3.BackColor = Color.Red;

button4.BackColor = Color.Red;

button5.BackColor = Color.Red;

button6.BackColor = Color.Red;

}

if (op == 3)

{

button1.BackColor = Color.Red;

button2.BackColor = Color.Red;

button3.BackColor = Color.Green;

button4.BackColor = Color.Red;

button5.BackColor = Color.Red;

button6.BackColor = Color.Red;

}

if (op == 4)

{

button1.BackColor = Color.Red;

button2.BackColor = Color.Red;

button3.BackColor = Color.Red;

button4.BackColor = Color.Green;

button5.BackColor = Color.Red;

button6.BackColor = Color.Red;

}

if (op == 5)

{

button1.BackColor = Color.Red;

button2.BackColor = Color.Red;

button3.BackColor = Color.Red;

button4.BackColor = Color.Red;

button5.BackColor = Color.Green;

button6.BackColor = Color.Red;

}

if (op == 6)

{

button1.BackColor = Color.Red;

button2.BackColor = Color.Red;

button3.BackColor = Color.Red;

button4.BackColor = Color.Red;

button5.BackColor = Color.Red;

button6.BackColor = Color.Green;

}

}

else

{

panel1.BackColor = Color.OrangeRed;

}}}

if (cbDetectMouthFile.Checked)

{

var mouths

= grayOfImageFrameCapturedFromCamera.DetectHaarCascade

(mouthHaar,

5,

3,

HAAR\_DETECTION\_TYPE.DO\_CANNY\_PRUNING,

newSize(1, 1))[0];

foreach (var mouth in mouths)

{

imageFrameCapturedFromCamera.Draw(mouth.rect, newBgr(Color.Black), 3);

intmouthLableX = mouth.rect.X;

intmouthLableY = mouth.rect.Y;

Point p = newPoint(mouthLableX, mouthLableY);

String m = "Mouth"; imageFrameCapturedFromCamera.Draw(m, refmouthFont, p, newBgr(Color.Black));

}

}

webCameraImageBoxRGB.Image = imageFrameCapturedFromCamera;

}}

#endregion

}

privatevoidbtnStart\_Click(object sender, EventArgs e)

{

if (capturedCaptureFromWebcam == null)

{

try

capturedCaptureFromWebcam = newCapture();

}

catch (NullReferenceExceptionexcpt)

{

MessageBox.Show(excpt.Message);

}

}

if (capturedCaptureFromWebcam != null)

{

if (isCaptureInProgress)

{

btnStart.Text = "Start Webcam";

Application.Idle -= DetectOrgansInFace;

}

else

{

btnStart.Text = "Stop Webcam";

Application.Idle += DetectOrgansInFace;

}

isCaptureInProgress = !isCaptureInProgress;

}

}

privatevoiddisposeDataFromTheMemory()

{

if (capturedCaptureFromWebcam != null)

capturedCaptureFromWebcam.Dispose();

}

privatevoidCameraCapture\_Load(object sender, EventArgs e)

{

synthesizer.Volume = 100;

synthesizer.Rate = -2;

synthesizer.SelectVoiceByHints(VoiceGender.Female, VoiceAge.Adult);

btnStart.Hide();

cbDetectEyeFile.Hide();

cbDetectFaceFile.Hide();

cbDetectMouthFile.Hide();

label2.Hide();

label1.Hide();

richTextBox1.Hide();

cbDetectFaceFile.Checked = true;

btnStart\_Click(null, null);

faceHaar = newHaarCascade("haarcascade\_frontalface\_alt\_tree.xml");

eyeHaar = newHaarCascade("haarcascade\_eye.xml");

mouthHaar = newHaarCascade("haarcascade\_Mouth.xml");

}

publicstaticvoidDetectSkin(BitmaporiginalImagePassed, refBitmapmodifiedImageWithSkin)

{

Graphics g = Graphics.FromImage(originalImagePassed);

ArrayList points = newArrayList();

for (inti = 0; i<originalImagePassed.Width; i++)

{

for (int j = 0; j <originalImagePassed.Height; j++)

{

ColorcolorOfPixel\_ij = modifiedImageWithSkin.GetPixel(i, j);

double I = (Math.Log(colorOfPixel\_ij.R) + Math.Log(colorOfPixel\_ij.G) + Math.Log(colorOfPixel\_ij.B)) / 3;

doubleRg = Math.Log(colorOfPixel\_ij.R) - Math.Log(colorOfPixel\_ij.G);

double By = Math.Log(colorOfPixel\_ij.B) - (Math.Log(colorOfPixel\_ij.G) + Math.Log(colorOfPixel\_ij.R)) / 2;

double hue = Math.Atan2(Rg, By) \* (180 / Math.PI);

if (I <= 5 && (hue >= 4 && hue <= 255))

{

points.Add(newPoint(i, j));

}

else

{

modifiedImageWithSkin.SetPixel(i, j, Color.Black);

}}}}

privatestaticvoid drawLineFromNode1ToNode2(Graphics g, Point p1, Point p2)

{

g.DrawLine(Pens.White, p1, p2);

}

privatestaticvoidSortPoints(refArrayListarrayOfPoints)

{

for (inti = 1; i<arrayOfPoints.Count; i++)

{

PointthisPoint = (Point)arrayOfPoints[i];

PointlastPoint = (Point)arrayOfPoints[i - 1];

if (thisPoint.X<lastPoint.X&&thisPoint.Y<lastPoint.Y)

{

}

else

{

swap(refarrayOfPoints, i - 1, i);

}

}

}

privatestaticvoid swap(refArrayListarrayOfPoints, inti, int j)

{

Point temp;

PointpointI = (Point)arrayOfPoints[i];

PointPointJ = (Point)arrayOfPoints[j];

temp = pointI;

pointI = PointJ;

PointJ = temp;

}

privatestaticintmaxOfRGB(int red, int green, int blue)

{

if (red > green && red > blue)

return red;

elseif (green > red && green > blue)

return green;

else

return blue;

}

privatestaticintminOfRGB(int red, int green, int blue)

{

if (red < green && red < blue)

return red;

elseif (green < red && green < blue)

return green;

else

return blue;

}

privatevoidbtnSkinDetection\_Click(object sender, EventArgs e)

{}

publicstaticBitmapScaleImage(ImageimagePassed, intmaxWidth, intmaxHeight)

{

doubleratioX = (double)maxWidth / imagePassed.Width;

doubleratioY = (double)maxHeight / imagePassed.Height;

double ratio = Math.Min(ratioX, ratioY);

intnewWidth = (int)(imagePassed.Width \* ratio);

intnewHeight = (int)(imagePassed.Height \* ratio);

BitmapscaledImage = newBitmap(newWidth, newHeight);

Graphics.FromImage(scaledImage).DrawImage(imagePassed, 0, 0, newWidth, newHeight);

returnscaledImage;

}

privatevoid timer1\_Tick(object sender, EventArgs e)

{}

privatevoid button6\_Click(object sender, EventArgs e)

{}

privatevoid button4\_Click(object sender, EventArgs e)

{}

privatevoid button5\_Click(object sender, EventArgs e)

{}

privatevoidwebCameraImageBoxRGB\_Click(object sender, EventArgs e)

{}

privatevoid button2\_Click(object sender, EventArgs e)

{}

privatevoid button3\_Click(object sender, EventArgs e)

{}

privatevoid button1\_Click(object sender, EventArgs e)

{}

privatevoid panel1\_Paint(object sender, PaintEventArgs e)

{ }}}

## RESULTS AND DISCUSSIONS

System development for detection and control of wink Tested using the 2.4 GHz IntelCoreQuad CPU Processor in Logitech USB Sequence. Webcam QuickC am 9000 Pro Image size entered. The sequence was equal to 320 \* 240 pixels test from the system took place in a room lighten by 3 fluorescents Lamps and natural light from a single window,i was a person sitting in front of the screen, where instructions for they were touching ask everyone to blink 40 times (20 long blinks and 20 short blinks, alternately). The USB camera is installed at the top of the screen, approx. at 50 cm from the person's face revealing of winking was done in real time with average speed 30 fps. Two types of errors were identified: the wrong discovery(The system detected a Wink when it did not exist) and lost winking (the current wink which was not discovered by the system). Potential distribution of the output is displayed to detect the wink in the form of graphs in Figure 5, 1Blinks is referred to correctly as it is (TP), False Positives are referred to as False Posi Fig.5.1

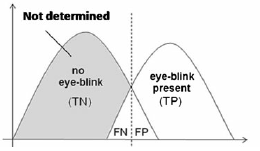


Fig. 5.1 Possible distribution of the eye-blink detector output

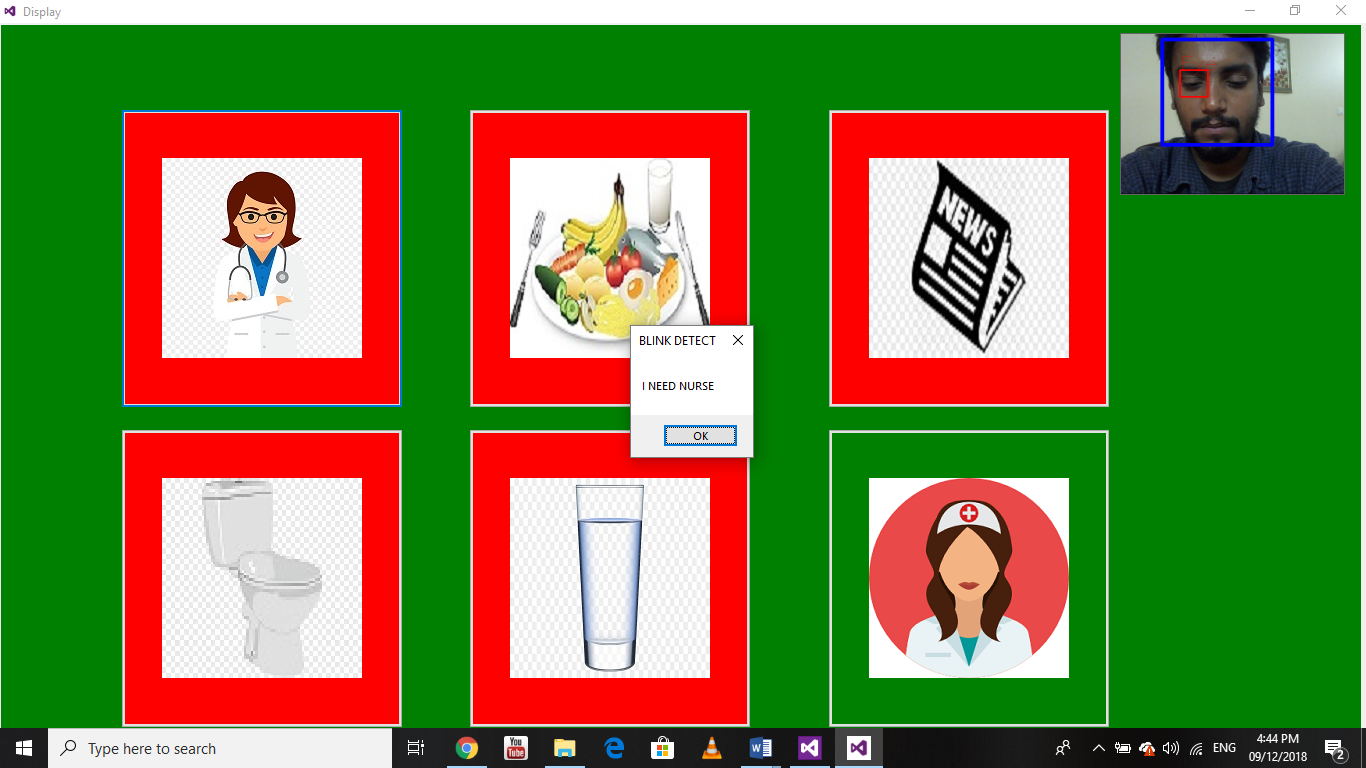


Figure no 5.2 command pass when double blink

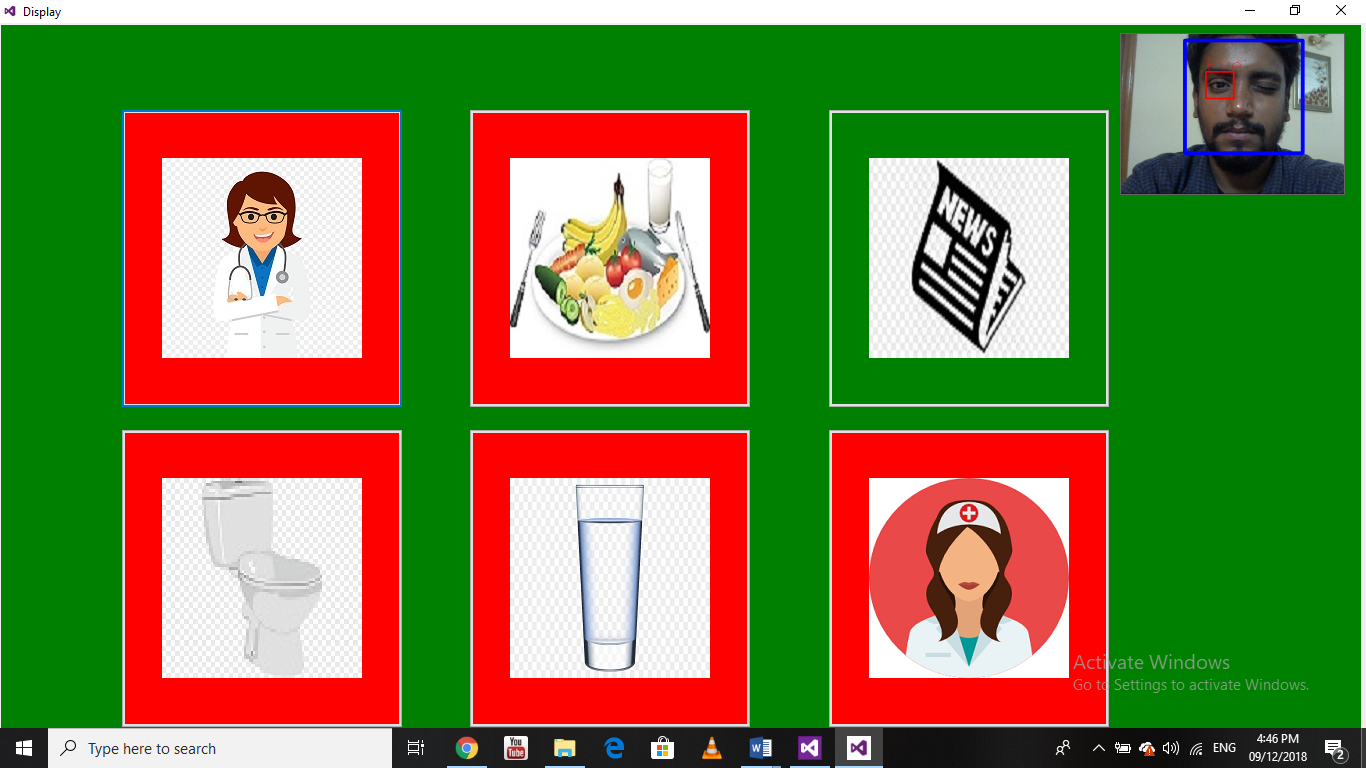


Figure no 5.3 moving left or right by eye wink

## CONCLUSION AND RECOMMENDATIONS

Automatic detection algorithm for voluntary winks, i was working in the development of the user interface. Applications written using C#,optical

Study Library and EmguCV. The system was built from rack components: Internet camera and consumer PC For best performance of system, and distance between camera and user Head should not be higher than 150 cm. System Configuration

It is shown in Figure 6.1.

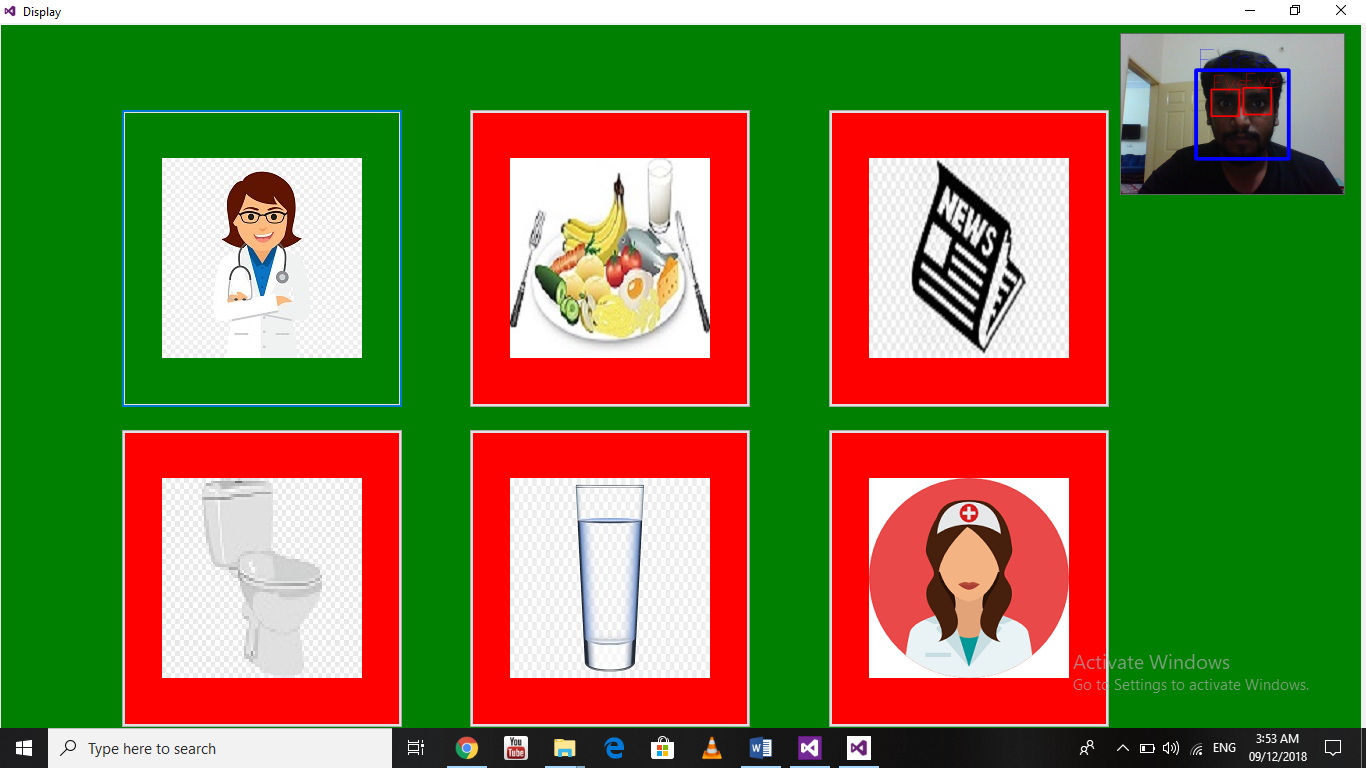


Fig. 6.1 Application Interface

The suggested interface is triggered by optional tabs the average duration of spontaneous winks is equals about 100ms. Detected winks are described as short (spontaneous) and long (voluntary) in order to avoid errors caused by the interpretation, spontaneous eye is a voluntary, flashing eye more than 200 milliseconds is considered "control". Some are interpreted as orders for the systemto

Example diagram to open the eyes in time with long and a short blink.

The proposed interface, designed for Windows operating system by us has the following features:

* Calling Doctor
* Need Food
* Want to read News
* Need to go to toilet
* Need water
* Need a nurse

The main items for a disabled patients are calling water, need of a drinking water, a need of a nurse, want to read a News and need to go to toilet all these operation can be performed with the help of winks, as in this application winks accuracy is approx... 83% and it is easily usable application.

There are lots of devices for eye gazing e.g. Tobii have been made for communication, interaction and learning purposes which are far expensive.

Here we develop a webcam based application which uses Haar classifier/Algorithm

For winkdetection and we recommend this as it is far cheaper than tobii devices and can be installed in a resberi-pie and is much efficient and which help the physically disabled patient to communicate with the doctor and around the people easily.

## REFERENCES.

[1] Alex Poole and Linden J. Ball, “Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects,” in Encyclopedia of Human Computer Interaction (30 December 2005) Key: citeulike:3431568, 2006, pp. 211-219.

[2] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, “Non-contact Eye Gaze Tracking System by Mapping of Corneal Reflections,” in Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02), 2002, pp. 94-99.

[3] Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena Lpez, “System for assisted mobility using eye movements based on electrooculography,” IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, vol. 10, no. 4, pp. 209-217, DECEMBER 2002.

[4] H. Singh and J. Singh, “A Review on Electrooculography,” International Journal of Advanced Engineering Technology, vol. III, no. IV, 2012.

[5] K. Irie, B. A. Wilson, and R. D. Jones, “A laser-based eye-tracking system,” Behavior Research Methods, Instruments, & Computers, vol. 34, no. 4, pp. 561-572, 2002.

[6] P Ballard and George C. Stockman, “Computer operation via face orientation,” in Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, 1992, pp. 407-410.

[7] T. Horprasert, Y. Yacoob, and L.S. Davis, “Computing 3-D head orientation from a monocular image sequence,” in Second International Conference on Automatic Face and Gesture Recognition, 1996, pp. 242- 247.

[8] K. Arai and M. Yamaura, “Computer Input with Human Eyes-Only Using Two Purkinje Images Which Works in a Real-Time Basis without Calibration,” CSC Journals, vol. 1, no. 3, pp. 71-82, 2010.

[9] D. Back, “Neural Network Gaze Tracking using Web Camera.,” Linkping University, MS Thesis 2005.

[10] R. Gonzalez and R. Woods, Digital Image Processing, 3rd ed.: Pearson Education, 2009.

[11] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in COMPUTER VISION AND PATTERN RECOGNITION, 2001.

[12] Fabian Timm and Erhardt Barth. Accurate eye centre localisation by means of

gradients. In Leonid Mestetskiy and Jos´e Braz, editors, VISAPP, pages 125–130. SciTePress, 2011. ISBN 978-989-8425-47-8. URL <http://dblp.uni-trier.de/db/> conf/visapp/visapp2011.html#TimmB11.

[13] R. Kothari and J.L. Mitchell. Detection of eye locations in unconstrained visual

images. In Image Processing, 1996. Proceedings., International Conference on, volume 3, pages 519–522 vol.3, Sep 1996. doi: 10.1109/ICIP.1996.560546.

[14] Tobii. http://www.tobii.com. Accessed: 2015-05-20.

[15] The eye tribe. https://theeyetribe.com. Accessed: 2015-05-20.

[16] Interactive-minds - eye tracking solutions. <http://www.interactive-minds.com>. Accessed: 2015-05-20. Bibliography 43

[17] Smart eye. http://smarteye.se. Accessed: 2015-05-20.

[18] Android developer. http://developer.android.com. Accessed: 2015-05-06.

[19] Opencv. http://opencv.org. Accessed: 2015-05-06.

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple

features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511–I–518 vol.1, 2001. doi: 10.1109/CVPR.2001.990517.

[21] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object

detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900–I–903 vol.1, 2002. doi: 10.1109/ICIP.2002.1038171.

[22] Opencv - object detection. http://docs.opencv.org/modules/imgproc/doc/

object\_detection.html. Accessed: 2015-05-20.

[23] Jiangang Wang, E. Sung, and R. Venkateswarlu. Eye gaze estimation from a single image of one eye. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 136–143 vol.1, Oct 2003. doi: 10.1109/ICCV.2003.1238328.

[24]H. Hua, P. Krishnaswamy, and J. P. Rolland, “Video-based eyetracking methods and algorithms in head-mounted displays,” *Optics Express*, vol. 1, no. 10, pp. 4328-4350, 2006

[25]Mr. Shivhari Eknathrao Lokhande Sagar D. Dharkar   
“EYE GAZE TECHNOLOGY”

[26]P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in *COMPUTER VISION AND PATTERN RECOGNITION*, 2001.

[27] Assit. Prof. Aree A. Mohammed and MSc. Student Shereen A. Anwer Computer Science Department College of Science “Efficient Eye Blink Detection Method for disabled-helping domain ”

[28] T.Morris P.Blenkhorn Farhan Zaidi “Blink Detection for Real-Time Eye Tracking ”