FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON

A PROJECT REPORT

Submitted by

MOHAMMED AL FAHAD (1914026)

PREM PRAKASH S (1914035)

KAVINRAJ V (1914507)

DEEPA V (1914701)

in partial fulfilment of the requirements for the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

K.S.R. COLLEGE OF ENGINEERING

(Autonomous)

TIRUCHENGODE – 637215

ANNA UNIVERSITY : : CHENNAI 600025

APRIL 2023

K.S.R. COLLEGE OF ENGINEERING

(Autonomous)

TIRUCHENGODE – 637215

ANNA UNIVERSITY : : CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report "**FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON"** is the bonafide work of **MOHAMMED AL FAHAD (1914026), PREM PRAKASH S (1914035), KAVINRAJ V (1914507), DEEPA V (1914701)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. S. RAMESH M.E., Ph.D.,	**Dr. S. RAMESH M.E., Ph.D.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor,	Professor,
Department of EEE,	Department of EEE,
K.S.R. College of Engineering,	K.S.R. College of Engineering,
Tiruchengode – 637 215.	Tiruchengode – 637 215.

Submitted for project Viva-Voce Examination held on _______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to place our sincere thanks and gratitude to our Founder **Theivathiru Lion Dr. K.S. RANGASAMY, MJF**, K.S.R. Educational Institutions and **Thiru R.SRINIVASAN B.B.M., MISTE.,** Chairman, K.S.R. Educational Institutions for providing us the needed facilities to do this project.

We express our heartfelt gratitude to our Principal **Dr.P.SENTHILKUMAR M.E., Ph.D.,** the man behind the making of thousands of professionals, for his valuable guidance and for motivating us to get success in life.

We like to express our sincere thanks to our Project supervisor and Head of the Department of Electrical and Electronics Engineering, **Dr.S.RAMESH M.E., Ph.D.,** for the guidance and support.

We would like to express our sincere gratitude to our mentors, all the faculty and staff members of Electrical and Electronics Engineering Department for their valuable support, towards the completion of this project.

MOHAMMED AL FAHAD

PREM PRAKASH S

KAVINRAJ V

DEEPA V

DEPARTMENT OF
ELECTRICAL AND ELECTRONICS ENGINEERING
K.S.R. COLLEGE OF ENGINEERING
(Autonomous)
TIRUCHENGODE - 637215
APRIL 2023

DECLARATION

We affirm that the project report titled being "**FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON"** submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidates.

MOHAMMED AL FAHAD
(1914026)

PREM PRAKASH S
(1914035)

KAVINRAJ V
(1914507)

DEEPA V
(1914701)

To verify that the declaration made by the above candidates is true to the best of my knowledge

Date: Name and Signature of the supervisor with seal

ABSTRACT

Physically disabled people are an important part of the society who has not yet received the same opportunities of inclusion as others in the Society. Therefore, it is necessary to develop easily accessible systems to achieve their inclusion within the new technologies. So, implementing a controlling system that enables them to use their arms without the help of another person is very helpful. The idea of facial movement control is of great use to not only the future of natural input but more importantly the handicapped and disabled. The objective of this project is to draw disabled people nearer to new technologies. In this project an assistive multimodal system is presented, which is aimed for the disabled people. The interaction between a user and a machine is performed by the algorithm that enables physically disabled individuals to control the robotic arm movement with the help of facial movements.

The novelty of the system is in terms of non invasiveness as earlier work mostly used glass based wearable trackers or head / face tracking, but no other earlier work reported use of webcam based eye tracking using python for controlling robotic arm by users. A camera capture's the facial movement and detects center position. Then the different variation on pupil position gets different command set for the robotic arm. Those signals will control the direction of the robotic arm to move up, down, left, right and stop. Also, allows the user to perform various actions such as choosing the specific joint to be controlled by winking the left or right eye. Moreover, the speed of the individual joints are also manually set the user's desired speed to increase the accuracy of the movement of the robotic arm. An automatic calibration is also implemented which increases the efficiency of system.

LIST OF CONTENTS

LIST OF CONTENTS

LIST OF CONTENTS

LIST OF CONTENTS

LISTOF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D	Three Dimensional
ADC	Analog to Digital Converter
BCI	Brain Computer Interface
COBOT	Collaborative Robots
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
DOF	Degrees of Freedom
EAR	Eye Aspect Ratio
FPS	Frames Per Second
GPU	Graphical Processing Unit
HMI	Human Machine Interface
IDE	Integrated Development Environment
MAR	Mouth Aspect Ratio
OPENCV	Open Source Computer Vision Library
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
SEC II	Smart Eye Communicator II
SSMI	Severe Speech and Motor Impairment
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Robotic arm is a programmable manipulator that comprises of linear and rotary joints to allow for controlled movements. It is widely used in various field such as industry, medical and military for many years due to its high repeatability, accuracy, and efficiency. The flexibility or dexterity of an articulated robotic arm is proportional to the number of axes of it. For the industrial articulated robotic arm, it ranges from different sizes depending on different application, for instance, the big heavy duty articulated arm performs automotive assembly while application such as electronics assembly is performed by smaller articulated arm.

What all these applications have in common is that the robotic arm is controlled mostly based on the input method via keyboard, mouse, or joystick. In some high tech applications even the use of hand gestures is also employed for the easy and dynamic movement of the various part of a robotic arm based on the movement of the hands. While this is not a problem for a healthy individual, this may be an insurmountable bound for people with limited freedom of movement of their limbs. In these cases, it would be preferable to use input methods which are based on more abilities of the region such as eye movements or head movement and map it to the movements of the robotic arm. The current technology is inadequate to be used by some people like ‘amputees’ as they do not have their hands to operate. Hence, the amputee and other physically challenged people will be able to operate if their eye movement or head movement is mapped to the robotic arm movement.

If the direction of the head movement is traced and direction towards which eye is looking at is also tracked, the movement will be mapped and the physically challenged people will be able to move the robotic arm. A facial movement based robotic arm will be a lot of use to disabled and an amputee as currently, this technology is not available at a large scale, only a few companies are trying to develop this technology and have made it available. To enable such substitute input methods a system was made which follows a low price approach to controlling of an arm based on facial movement. The head tracker is based on images recorded by a mutated webcam to acquire the head movements. These head movements are then graphed to servo motor movements that determine the various position of each individual joint of the robotic arm.

1.2 INTRODUCTION TO PYTHON

Python is a widely used general purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. It is a programming language that lets the user work quickly and integrate systems more efficiently. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages. Python supports multiple programming paradigms, including procedural, object oriented, and functional programming. It also has a dynamic type system and automatic memory management, which makes it easy to write and maintain code. Python has become one of the most popular programming languages and it is widely used in many industries and its versatility and ease of use make it an ideal choice for a wide range of projects.

1.3 INTRODUCTION TO OPENCV

Open Source Computer Vision (OpenCV) is a popular open source library for computer vision, machine learning, and image processing applications. It was first released in 2000 by Intel and is now maintained by the OpenCV Foundation. OpenCV provides a large collection of functions and algorithms for image and video analysis, including image filtering, feature detection, object recognition, and machine learning. It supports various programming languages, including C++, Python, Java, and MATLAB, and runs on multiple platforms, including Windows, Linux, and macOS.

1.4 INTRODUCTION TO PY FIRMATA

Py Firmata is a python interface for the firmata protocol. Firmata is an intermediate protocol that connects an embedded system to a host computer, and the protocol channel uses a serial port by default. It is intended to work with any host computer software package. Right now, there is a matching object in a number of languages. It is easy to add objects for other software to use this protocol. Basically, it establishes a protocol for talking to the Arduino.

1.5 INTRODUCTION TO DLIB

Dlib is a modern programming toolkit containing machine learning algorithms and tools for creating complex software in programming languages to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dib’s open source licensing allows it to be used in any application, free of charge.

1.6 PROBLEM STATEMENT

People with severe speech and motor impairment (SSMI) often uses a technique called eye pointing to communicate with outside world. One of their parents, caretakers or teachers hold a printed board in front of them and by analysing their eye gaze manually, their intentions are interpreted. This technique is often error prone and time consuming and depends on a single caretaker.

1.7 MOTIVATION

Even after a decade of research in the field of eye gaze controlled robotic arm, a commercial solution that is both cost effective and efficient is yet to be proposed. Therefore, we aim to undertake a user centred design process to automate the process of eye gaze tracking by using a simple web camera that provides a human robot interaction for users with SSMI.

1.8 OBJECTIVE OF THE PROJECT

- The objective of the project is to automate the eye tracking process electronically by using commercially available tablet, computer, or laptop and without requiring any dedicated hardware for eye gaze tracking.
- By using the tracking data, the robotic arm will be manipulated based on the user’s input with the help of the python program.

1.9 ORGANIZATION OF THE REPORT

Chapter 1 deals with the introduction and describes the importance of the various components that are used in system for facial movement detection and the manipulation of the robotic arm.

Chapter 2 provides the details about the various literatures and improvement needed for the different authors are also be discussed.

Chapter 3 discusses about overall structure and the working of the proposed system in detail. The block diagram and the system architecture of the proposed system is also shown here.

Chapter 4 provides the description about the hardware setup of the proposed system in detail. It also provides the specifications and images of the various components that are used in the system.

Chapter 5 gives a detailed explanation about the various programs that are used in the software setup of the proposed system. The mathematical equations that are used in the system is also discussed in this chapter.

Chapter 6 shows the results of facial movement based robotic arm movement and the outcome photos. This section additionally comprises of the coding of the framework.

Chapter 7 gives the conclusion of the project. It says about the overview and importance of implementing the project. In this chapter, the possibilities for future research are also stated.

1.10 SUMMARY

This chapter provides an overview of facial movement based robotic arm movement using Arduino and Python and its objective is also discussed. A brief introduction of the rest of the chapters has been outlined. The next chapter examines the literature review of numerous studies.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

This chapter reviews the literature presented related to the facial movement based robotic arm control using Arduino and Python. The drawbacks which are having in the journal papers have been analysed and the way to overcome the problems are explained in detail. The scope of the project is also discussed in this chapter.

2.2 SURVEY ON FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON

V K Sharma et al., (2021) proposed a non invasive eye gaze controlled robotic manipulator system for people with SSMI, in which a video see through eye gaze controlled interface and a webcam based eye gaze estimation software was developed and then combined into a single system where users with SSMI will be able to control a robotic arm using only a webcam. [1] Many earlier research explored controlling robotic manipulator or wheelchair through eye gaze or head movement although many of them were not evaluated with more than 5 users with severe motor impairment. They worked with a set of users who are severely motor impaired, and the level of disability is more severe than participants recruited for earlier research works and also undertook two parallel activities on developing a webcam based eye gaze tracker that does not require any extra hardware with a standard laptop and developing a video see through human robot interface that does not require users to wear any eye tracking glass.

The video see through interface only used icons and thus could be used by different language speakers. The user studies demonstrated that users with SSMI undertook representative pick and drop tasks using the system. It was noted that the average task completion times were higher for webcam based gaze tracker than Commercial Off The Shelf (COTS) eye gaze tracker. A COTS eye tracker uses dedicated ASIC chip for video processing while a webcam based gaze tracker uses the computer processor, which is shared with other programs. It may also be noted that any high end Graphical Processing Unit (GPU) was not used for the reported trials. An earlier study undertook detailed analysis on visual search and fixation patterns on similar set of users and found that they also suffer from Nystagmus and cannot fixate attention in all regions of screen with equal ease as their able bodied counterpart. However, even considering these difficulties it may be noted that with practice, the task completion times are reduced further and will be nearly equal to COTS eye gaze tracker for a user interface with limited screen elements.

Jeevithashree et al., (2019) proposed two different systems on developing and configuring a webcam for eye gaze tracking and then used the webcam based gaze trackers in a gaze controlled interface for users SSMI. As part of the study, an online quiz application was implemented with only four screen elements. The study showed that able bodied users could use the webcam based gaze tracker to operate the quiz applications, but users with SSMI were able to operate the system significantly faster with a low cost infrared based commercial eye gaze tracker. The viola jones landmark based eye gaze tracker and webgazer.js for users with SSMI was also compared but it was reported with an average selection times more than 10 secs, which makes it unusable for practical purpose. [2]

Rosebrock A (2019) proposed an eye gaze tracking system built to operate the communication system. The processes which were used in this system are the face and left eye area of user which was captured by camera on android device and the features were be detected by Haar Cascade and from left eye's region, it looked for eye center position as user pointer's reference in interface menus using moments. User could choose the activity menu by closing their eye in time more than 250ms. The output of this system was an audio that represent activity menu. The system had percentage success rate of left gaze tracking with 80 %, straight gaze tracking with 90 %, and right gaze tracking with 80 %. However, the paper did not report any evaluation on pointing and selection times for users with SSMI and accuracy was only measured in terms of eye detection. [3]

Baltrusaitis et al., (2018) designed a wearable system which used Emotiv Epoc recording headset along with custom built eye tracker. [4] The pupil centre was assumed to be an ellipse and was estimated from the binarized image. RANSAC algorithm was used to remove outliers from extracted points. Furthermore, the system interpolated gaze point using second order polynomial. Eye movement was mapped to cursor movement and Brain Computer Interface (BCI) was used for selection task. The system was evaluated with four different interface protocols using Fitts' Law task with 9 able bodied users. There are also a few commercial webcam based gaze trackers like Web gazer or xLabsgaze, but they are mainly advertised for recoding browsing behaviour of web users.

Biswas et.al., (2018) proposed the Smart Eye Communicator II (SEC II), an evolution of a previously presented algorithm to detect the pupil center and the user's gaze direction in real time, using a low resolution webcam and a conventional computer without a need for calibration. [5]

In SEC II, a face aligner, which gets a canonical face alignment based on translation, scale, and rotation, has been added to the system. Likewise, strategies using eye coordinates were implemented to find the dominant eye. By implementing these new approaches, the algorithm achieved 86% accuracy, even under variable and non uniform environmental conditions.

Koch et.al., (2017) proposed a methodology for real time eye gaze tracking using a standard webcam without the need for hardware modification or special placement. [6] An artificial neural network was employed to estimate the location of the user's gaze based on an image of the user's eye, mimicking the way that humans determine where another person is looking. Accuracy measurements and usability experiments were performed using a laptop computer with a webcam built into the screen. The results show this approach to be promising for the development of usable eye tracking systems using standard webcams, particularly those built into many laptop computers. But they reported problem in extrapolation about training the network while detecting eye gaze for one of 50 random points on screen and also had drawbacks such as slow algorithms which used a lot of processing resources and did not detect face, rather directly detected eyes, and tested for only five positions (Right, Left, Straight, Up and Down) on screen.

Alsharif S et.al., (2016) proposed the SearchGazer system that extends webgazer system with a regression model that maps eye features to gaze locations and search page elements during user interactions. [7] It was a pure JavaScript library that infers the gaze behaviour of searchers in real time. The eye tracking model self calibrates by watching searchers interact with the search pages and trains a mapping of eye features to gaze locations and search page elements on the screen.

Contrary to typical eye tracking studies in information retrieval, this approach did not require the purchase of any additional specialized equipment and could be done remotely in a user's natural environment, leading to cheaper and easier visual attention studies.

Bremner et.al., (2016) proposed a technique to develop a non intrusive interface of providing non contact head and eye based control of computer systems for people with motor difficulties. Such an interface was proposed to replace a traditional interface like a normal mouse thus helping the severely disabled to use the computer just as a normal individual would. It used human iris tracking and blink detection for this purpose. Considering constant lighting conditions, it extracts the between the eyes features followed by eye localization using Haar like features classifier. Circular Hough transform was used to track the iris movement at real time under varying lighting conditions. Accurate head and iris tracking results were obtained at a processing rate of 30 Frames Per Second (FPS) in more than 90% cases with a low false positive blink detection rate of 0.01%. [8]

Dziemian et.al., (2016) proposed a method that detects the iris region from low resolution eye region images by its intensity values rather than the shape, ensuring that this region could also be detected at different angles of rotation and under partial occlusion by the eyelids. [9] Following the calculation of the point of regard from the estimated iris center coordinates, a number of Kalman filters were used to improve upon the noisy point of regard estimates to smoothen the trajectory of the mouse cursor on the monitor screen.

Khonglah et.al., (2015) proposed a real time eye gaze estimation system by using a general low resolution webcam, which could estimate eye gaze accurately without expensive or specific equipment. [10]

An illuminance filtering approach was designed to remove the influence from light changes so that the eyes are detected correctly from the low resolution webcam video frames. A hybrid model combining the position criterion and an angle based eye detection strategy were also derived to locate the eyes accurately and efficiently. In the eye gaze estimation stage, a Fourier Descriptor was employed to describe the appearance based features of eyes compactly. The determination of eye gaze position was then carried out by the Support Vector Machine. The system used SVM to classify the eye images to get the gaze points at one of the 9 gaze locations.

Hansen et.al., (2014) introduced GazeCapture that used a deep convolutional neural network trained using gaze capture data set for predicting eye gaze and detect attention in one of three monitors but not used as a cursor control algorithm to operate a single display. It contained data from over 1450 people consisting of almost 2.5M frames. [11]

Dostal et.al., (2013) introduced a novel gaze gesture based Human Robot Interface to control a robot arm with seven Degrees of Freedom (DOF) using the eye movement as the sole input. [12] To make the best use of the two dimensional information provided by the eye tracking glasses, different control modes were defined to translate, rotate, open, or close the robot gripper. Gaze gestures and state of the eyes are used to switch between different control modes. A user study was conducted to test the usability of the proposed interface configured eye gaze movement.

The performance of the system was evaluated with 10 participants including one person with motor impairment for a block rearrangement task. But it had drawbacks such as use of costly eye tracking device.

Bannat et.al., (2011) introduced performance improvements to an eye tracking system that was previously developed and was used it to explore if that approximation is appropriate. [13] More precisely, they compared the impact of the use of eye or head based gaze estimation in a human robot interaction experiment with the iCub robot and reported significant reduction in task completion time and increase in subjective preference for eye gaze tracking system compared to head tracking system.

San Agustin et.al., (2010) proposed a novel approach for multimodal interactions between humans and industrial robots. [14] The application scenario was situated in a factory, where a human worker is supported by a robot to accomplish a given hybrid assembly scenario, that covers manual and automated assembly steps. The robot was acting as an assistant as well as a fully autonomous assembly unit. For interacting with the presented system, the human was able to give his commands via three different input modalities (speech, gaze and buttons).

Margrit Betke et.al., (2009) evaluated an ethnographic study conducted in an elementary school where 40 children interacted with a social robot capable of recognising and responding empathically to some of the children's affective states. The empathic behaviour of an autonomous social robot was evaluated it and it was reported that empathy facilitates the interaction and affects positively the perception of robot. [15]

Bonneau et.al., (2009) investigated how robot mediation affects the way the personality of the operator is perceived. [16]

More specifically, it aimed to investigate if judges of personality is consistent in assessing personality traits that agrees with one another or agree with operator’s self assessed personality and shift their perceptions to incorporate characteristics associated with the robot's appearance and effect of personality cues of a robotic tele avatar like gesture, speech, and appearance on its perception by the operator. They also reported that the perception and individual behaviour of operator is subjective and vary for individuals and concluded that any human robotic system should be designed considering the user background, perception, and behaviour.

Chen Y et.al., (2008) proposed a Human Machine Interface (HMI) based on gaze tracking proposed to control robot prostheses. [17] The Robot manipulators held a strong similarity with arm prosthetics and used a 7 DOF whole arm manipulator to test HMI in the execution of reaching and grasping tasks. It showed that the interface worked under different control strategies using several velocity profiles. The system was tested by ten subjects with encouraging results. The performance of the 7 DOF robot manipulator was analysed in order to determine the suitability of its application in the development of this project.

Dautenhahn et.al., (2005) was interested in how gaze interaction is applied to robotic systems, to provide new opportunities for people with physical challenges. [18] However, since only a few studies have implemented gaze interaction into a telepresence robot, and it was still unclear how gaze interaction within these robotic systems impacts users and how to improve the systems. So, a new system was introduced to operate a remotely located robot and reported user studies with able–bodied participants.

Kim et.al., (2001) used electrooculography to detect eye gaze movements and used it to control robotic movement in 8 directions. [19] This technique detected the movement of the eyes by measuring the difference of potential between the cornea and the retina by placing electrodes around the ocular area. The processing algorithm was developed to obtain the position of the eye and the blink of the user. The output of the processing algorithm offered, apart from the direction, four different values zero to three to control the velocity of the robot arm according to how much the user is looking in one direction.

Fitts P.M. (2001) proposed an intuitive eye tracking controlled robot arm operating in 3 dimensional space based on the user's gaze target point that enables tele writing and drawing by using an eye gaze controlled interface and a screen mounted low cost eye tracker to move a prosthetic arm but did not use video see through display and evaluated the system with only one able bodied user. [20]

2.3 SCOPE OF THE PROJECT

The scope of the project is to develop a user friendly and cost effective robotic arm that is controlled by using an Arduino and Python. This project aims to provide a multimodal robotic arm control technique by using natural facial movements for people with various disabilities.

2.4 SUMMARY

This chapter reviews the literature presented related to the Facial movement based robotic arm control using Arduino and Python. The drawbacks which are present in the journal papers were analysed and the way to overcome the problems are also explained in detail. The scope of the project is also discussed in this chapter.

CHAPTER 3

SYSTEM DESCRIPTION

3.1 INTRODUCTION

This chapter reviews the detailed description of the overall working of the system used in the Facial movement based robotic arm control using Arduino and Python by using a block diagram. This chapter gives a clear note of the Existing and Proposed systems also provides the details of the overall algorithm used and the system architecture.

3.2 EXISTING SYSTEM

People with SSMI often uses a technique called eye pointing to communicate with outside world. Figure 3.1 shows the printed board that either parents, caretakers or teachers hold in front of them and by analysing their eye gaze manually, their intentions are interpreted. This technique is often error prone and time consuming and depends on a single caretaker.

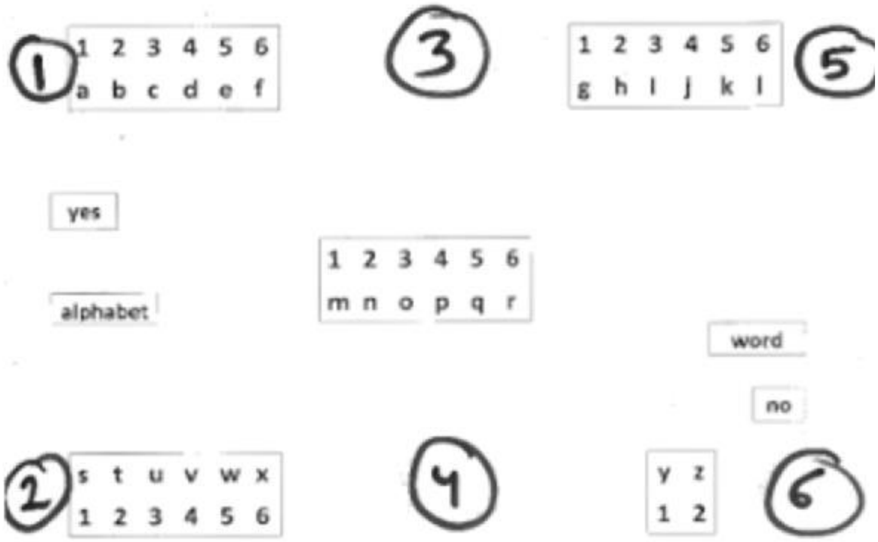


Figure 3.1 Non electronic eye pointing chart

To overcome the limitations for non electronic solutions, electronics based eye tracking is used. Eye tracking is the process of measuring either the point of gaze where one is looking or the motion of an eye relative to the head. Eye tracking is traditionally used for analysing visual perception, eye gaze movement and making visual perception models. In recent times, eye gaze has also been used to directly control a graphical user interface. Eye gaze controlled interfaces have been used for people with SSMI, who cannot use existing computer peripherals like mice, touchpads, or keyboards. These interfaces have also recently been explored for able bodied users in situational impairments like drivers and pilots, whose hands are engaged with the primary task of driving. Commercial eye gaze trackers have the advantage of higher accuracy but those need to be separately bought and configured for individual computers. Webcam based eye gaze trackers are far less accurate than infrared based commercial ones, but if it is found useful even for a limited set of applications, that are used without the need of buying or configuring any dedicated hardware.

3.2.1 Existing Block Diagram

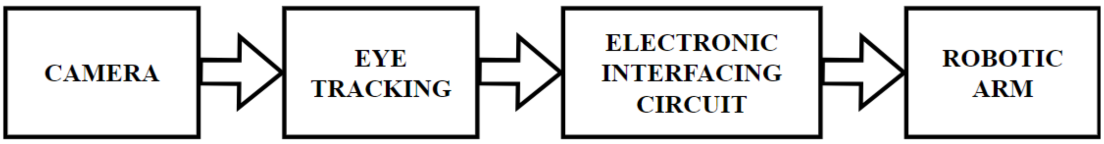


Figure 3.2 Existing block diagram

The existing system was developed by tracking V K Sharma et.al., (2021) to automate the process electronically by using commercially available computer or laptop and without requiring any dedicated hardware for eye gaze. The figure 3.2 shows the block diagram of the existing system.

The system used a user centred design process and separately evaluated the webcam based gaze tracker and the video see through based human robot interaction involving users with SSMI. It also reported a user study on manipulating a robotic arm with webcam based eye gaze tracker.

In that system the gaze trackers captured the live video of the face of the user and processed the video to estimate eye gaze direction. A transparent user interface was rendered on a video see through display and user could operate the display by dwelling eye gaze on different screen elements. As the user selects a screen element, a command will be sent to an electronic interfacing circuit that will instruct the robotic arm to undertake an action fulfilling the user's intention. It also presented user studies involving both able bodied users and people with SSMI while developing bespoke eye tracker for comparing different implementations of webcam based eye gaze tracker with users with SSMI and also to evaluate the object manipulation tasks involving users with SSMI using a robotic arm.

3.2.2 Drawbacks of Existing System

- **Mounting devices:** These systems needed a mounted device like lasers or cameras on the user which became tedious
- **Biometric identification:** The system used biometric identification for which the users had to register themselves before using the system. It wasn't open for all which has been rectified by the proposed application
- **Complex algorithms:** The previous systems used many complex algorithms that needed a lot of calculations to be done depending on various markers

3.3 PROPOSED SYSTEM

In our proposed system the robotic arm movement of is controlled by eye movement using OPENCV. Camera detects the Eyeball movement which is processed in OPENCV. By this the robotic arm is controlled and the user has to sit in front of the display screen of private computer or pc, a specialised video camera established above the screen to study the consumer's eyes.

3.3.1 Proposed Block Diagram

The program constantly analyses the video and determines where the user is looking at the display screen. To pick out any action, the user seems at the direction for an exact period and to perform a particular action. The system inputs the video frames from the user. The video is pre processed for enhancement. In this process the noise and the blurriness of the video is handled, and the video is divided into some frames and the face detection is carried out using the object detection algorithm. In figure 3.3 shows the proposed block diagram of the proposed system.

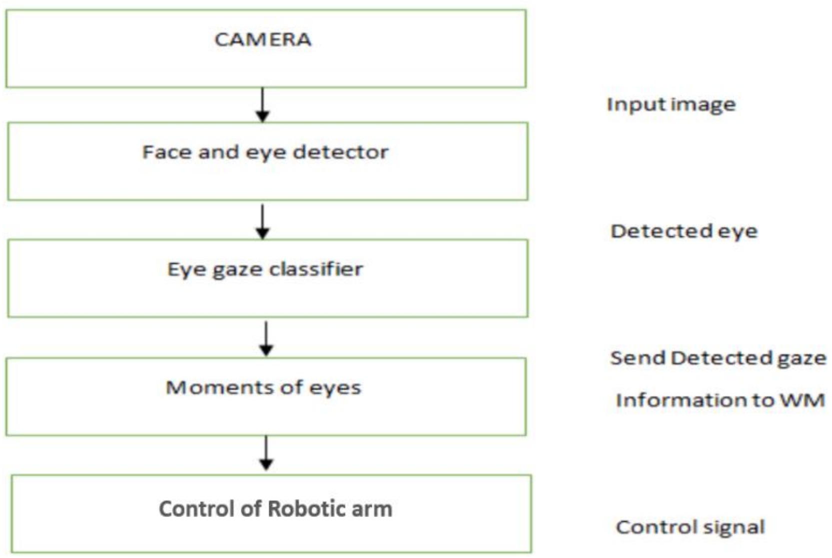


Figure 3.3 Proposed block diagram

3.4 PROPOSED SYSTEM ALGORITHIM

1) Camera receives the input from the face
2) After receiving these streaming videos from the camera, it is broken into frames
3) After receiving frames, it will check for lighting conditions because cameras require sufficient lights from external sources otherwise error message will be displayed on the screen
4) If the lighting conditions are satisfied facial landmarks are placed over the image with the help of the dlib program to aid in the facial recognition algorithm
5) Images (frames) from the input source focusing the eye are analysed for Iris detection (center of eye)
6) Finding the center of the iris. This step forms a major part of the algorithm. The algorithm first finds an 'eye window' – a rough estimate of the eye using voila jones algorithm
7) The second step is to find the exact position of the iris within the eye window using Houghman Circle Detection Algorithm
8) Mapping the iris to a point in the video from the scene camera. Using pre determined calibration points, the position of the iris is mapped to a position on the screen
9) After this, a mid point is calculated by taking the mean of left and right eye center point
10) Later, the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) is calculated
11) Finally, the facial movement data is converted into servo motor angles and fed to the pyfirmata program for the control of the robotic arm using Arduino

3.5 PROPOSED SYSTEM ARCHITECTURE

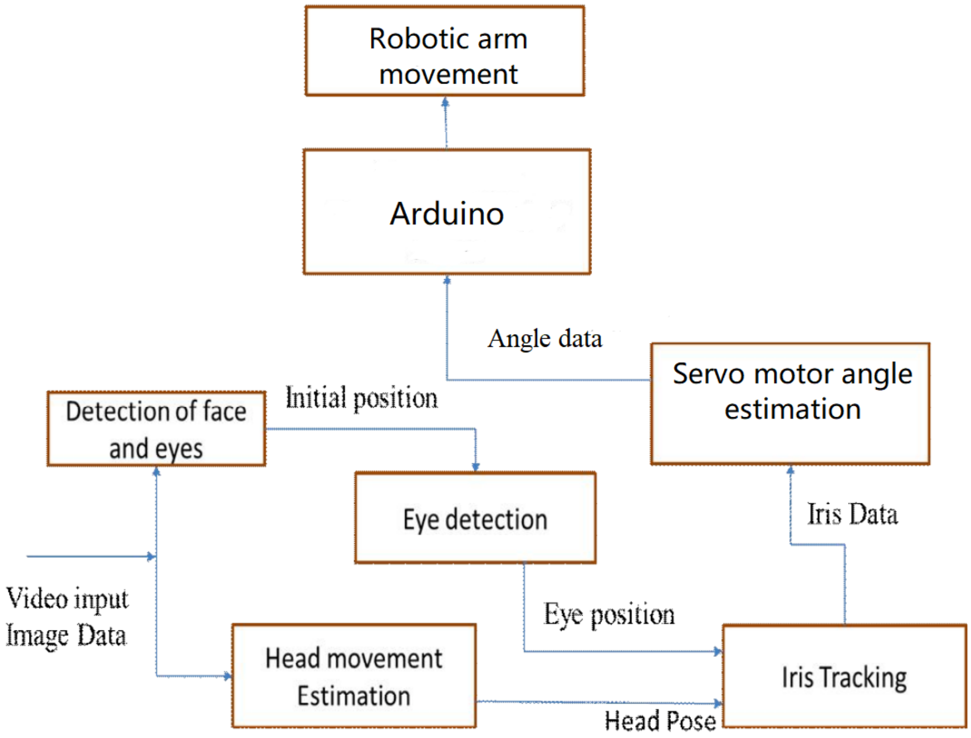


Figure 3.4 Proposed system architecture

The system is divided into three components:

1) Face detection and tracking component
2) Eye tracking component
3) Gaze to robotic arm movement component

The proposed system architecture is given in the figure 3.4. It uses a camera with 20 megapixels (480 pixels interpolated 20M pixels still image resolution, interpolated 2.1M pixels video resolution) to capture the images of the user for iris tracking and gaze estimation.

The video captured by webcam has framerate around 30 FPS, at every 1/30th of a second a frame will be processed. Before the features of the frame are detected the frame goes through a set of processes and then plotted to the cursor.

This process interminably occurs as a part of a loop for every frame and the first image is used for initial face location and eye detection. If any one of these detection procedures fails, then go to the next frame and restart the above detection processes. If iris tracking fails, the processes of face location and eye detection restart on the present image. These procedures continue until there are no more frames. Once this process is over, with the help of the dlib program facial landmarks or point are placed over the face to aid the facial tracking component.

Once the points or landmarks are placed over the face in the captured image by using Dlib, this data is processed by Python and the facial movement data is converted into servo motor angles and this data is fed into the pyfimata program to transfer the data to the Arduino which then control the servo motor and moves the robotic arm as per the users desire and movement.

3.6 SUMMARY

This chapter provided the detailed description of the existing system and its drawback, and the proposed system and it’s working with the help of a block diagram and also provided an explanation about the system architecture and the algorithm used for the functioning of the system.

CHAPTER 4

HARDWARE DESCRIPTION

4.1 INTRODUCTION

This chapter reviews the detailed description of the various hardware components used in the Facial movement based robotic arm control using Arduino and Python the technology behind that with the specification of those module components. This chapter also shows about the images of the components used in the device’s system.

4.2 ARDUINO UNO

Arduino UNO is a microcontroller board based on the ATmega328P chip. It is one of the most popular microcontroller boards due to its ease of use, versatility, and low cost. It is open source, which means that anyone will be able to access and modify the board's schematics, software, and documentation. The board comes with a set of input/output pins, digital and analog, that are used to connect the board to various sensors, actuators, and other components. Figure 4.1 shows the image the Arduino uno.

Figure 4.1 Arduino UNO

The board is used in a wide range of applications, from simple hobby projects to complex industrial automation systems as the pins are also programmed to perform specific tasks, such as reading data from a sensors or sending data to a servo motor. Arduino UNO is also compatible with a wide range of sensors and components, which makes it easy to prototype and test various ideas quickly. Furthermore, Arduino UNO has a Universal Serial Bus (USB) port that is used to upload code from a computer to the board and to communicate with it and also be programmed using a variety of programming languages, including C++, Python, and Scratch, which makes it accessible to both novice and advanced users so unlike traditional microcontrollers, which require advanced programming skills, Arduino is programmed using a simple and user friendly Integrated Development Environment (IDE). The IDE comes with a library of pre written code, which makes it easy for beginners to get started.

- Microcontroller - ATmega328P
- Clock speed - 16MHz
- Digital I/O pins - 14 (including 6 PWM pins)
- Analog input pins - 6
- Flash memory - 32KB (including 0.5KB used by bootloader)
- SRAM - 2KB
- EEPROM - 1KB
- USB interface - for programming and serial communication with a computer
- Power supply – It is powered via USB, external power supply or battery

4.2.1 Arduino Architecture

The Arduino architecture refers to the hardware and software components that make up the Arduino platform. Here are the main components of the Arduino architecture:

- Microcontroller - The Arduino board is based on a microcontroller, such as the ATmega328P or ATmega2560. The microcontroller is the main processing unit of the board and is responsible for executing the program instructions
- Clock - The Arduino board includes a clock crystal that generates a signal used to synchronize the operation of the microcontroller
- Power supply - The Arduino board is powered using a USB connection or an external power supply. The power supply is regulated to provide a stable voltage to the microcontroller and other components
- Input/output pins - The Arduino board includes digital and analog input/output pins are used to interface with sensors, actuators, and other components. The digital pins are used for binary input or output, while the analog pins are used to read analog signals from sensors
- Communication interfaces - The Arduino board includes serial communication interfaces, such as USB that is used to communicate with a computer or other devices. It may also include other communication interfaces for interfacing with sensors and other components
- Arduino IDE - The Arduino IDE is a software tool that allows the user to write, compile, and upload code to the Arduino board. It includes a code editor, compiler, and uploader to develop and test programs

The Arduino architecture is designed to be user friendly and easy to use, even for beginners. The combination of hardware and software components allows users to quickly prototype and develop electronics projects, without the need for advanced programming skills or knowledge of electronic circuits.

4.2.2 ATmega328 Microcontroller

The high performance, low power Atmel 8 bit RISC based microcontroller has 16 KB flash memory, 1KB SRAM, 512B EEPROM, an 8 channel/10 bit A/D converter and debug for on chip debugging. The typical pin diagram of the ATmega328 microcontroller is shown in the Figure 4.2. The device supports a throughput of 20 MIPS at 20 MHz and operates between 2.7 to 5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

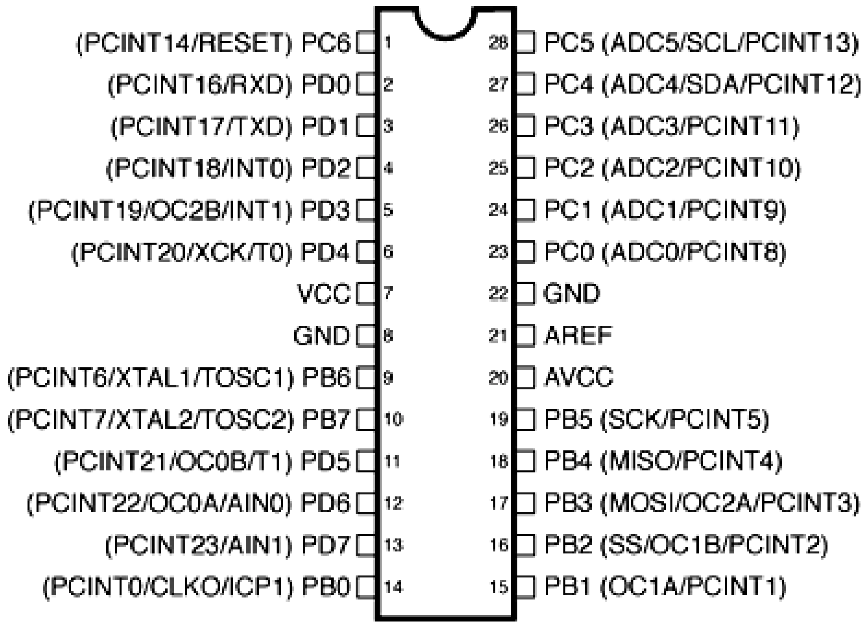


Figure 4.2 Pin diagram of ATmega32

4.2.3 Features of ATmega328

The ATmega328 microcontroller has numerous features that make it an ideal platform for a wide range of electronic applications. The Table 4.1 shows the features and specification of ATmega328 microcontroller.

Table 4.1: Features and specification of ATmega328

Feature	Specification
CPU	8 bit AVR
Number of Pins	28
Operating Voltage (V)	+1.8 V TO +5.5V
Number of programmable I/O lines	23
Communication Interface	Master/Slave SPI Serial Interface (17,18,19 PINS) Programmable Serial USART (2,3 PINS) Two wire Serial Interface (27,28 PINS)
JTAG Interface	Not available
ADC Module	6channels, 10 bit resolution ADC
Timer Module	Two 8 bit counters with Separate Prescaler and compare mode and One 16 bit counter with Separate modes.

Feature	Specification
Analog Comparators	1(12,13 PINS)
PWM channels	6
Program Memory Type	Flash
Internal Oscillator	8MHz Calibrated Internal Oscillator
External Oscillator	0 4MHz @ 1.8V to 5.5V 0-10MHz @ 2.7V to 5.5V 0-20MHz @ 4.5V to 5.5V
Program Memory or Flash memory	32Kbytes [10000 write/erase cycles]
CPU Speed	1MIPS for 1MHz
RAM	2Kbytes Internal SRAM
EEPROM	1Kbytes EEPROM
Watchdog Timer	Programmable Watchdog Timer with Separate On chip Oscillator
Program Lock	Yes
Power Save Modes	Six Modes [Idle, ADC Noise Reduction, Power save, Power down, Standby and Extended Standby]
Operating Temperature	-40°C to +105°C

4.2.4 ATmega328 Pin Configuration

The ATmega328 microcontroller has a total of 28 pins, which are used for a variety of purposes, including digital input/output, analog input, and serial communication. Understanding the pin description is essential for correctly interfacing with the microcontroller and using its features to their full potential. The table 4.2 shows the pin description and its function of ATmega328 microcontroller.

Table 4.2 ATmega328 pin description

Pin No.	Pin name	Description	Function
1	PC6	Reset	When this reset pin goes low the microcontroller & its program gets reset.
2	PD0	Digital Pin (RX)	Input pin for serial communication
3	PD1	Digital Pin (TX)	Output pin for serial communication
4	PD2	Digital Pin	Pin 4 is used as an external interrupt 0
5	PD3	Digital Pin (PWM)	Pin 5 is used as an external interrupt 1
6	PD4	Digital Pin	Pin 6 is used for external counter source Timer0

Pin No.	Pin name	Description	Function
7	Vcc	Positive Voltage	Positive supply
8	GND	Ground	Ground of the system
9	XTAL	Crystal Oscillator	This pin is connected to one pin of the crystal oscillator to provide external clock pulse to the chip
10	XTAL	Crystal Oscillator	This pin is connected to another pin of the crystal oscillator to provide external clock pulse to the chip
11	PD5	Digital Pin (PWM)	Pin 11 is used for external counter source Timer1
12	PD6	Digital Pin (PWM)	Positive Analog Comparator i/ps
13	PD7	Digital Pin	Negative Analog Comparator i/ps
14	PB0	Digital Pin	Counter or Timer input source
15	PB1	Digital Pin (PWM)	Counter or Timer Compare match A output

Pin No.	Pin name	Description	Function
16	PB2	Digital Pin (PWM)	This pin is act as a slave choice i/p.
17	PB3	Digital Pin (PWM)	This pin is used as a master data output and slave data input for SPI.
18	PB4	Digital Pin	This pin is act as a master clock input and slave clock output.
19	PB5	Digital Pin	This pin is act as a master clock output and slave clock input for SPI.
20	AVcc	Positive Voltage	Positive voltage for ADC (power)
21	AREF	Analog Reference	Analog Reference voltage for ADC (Analog to Digital Converter)
22	GND	Ground	Ground of the system
23	PC0	Analog Input	Analog input digital value channel 0

Pin No.	Pin name	Description	Function
24	PC1	Analog Input	Analog input digital value channel 1
25	PC2	Analog Input	Analog input digital value channel 2
26	PC3	Analog Input	Analog input digital value channel 3
27	PC4	Analog Input	Analog input digital value channel 4. This pin is also be used as serial interface connection for data.
28	PC5	Analog Input	Analog input digital value channel 5. This pin also used as serial interface clock line.

4.3 SERVO MOTOR

A servo motor is an electromechanical device that is commonly used in various industrial and commercial applications, such as robotics, automation, and manufacturing. It is a type of motor that is designed to accurately control the position, speed, and acceleration of a load. The basic components of a servo motor include a motor, a feedback device, and a controller. The motor is responsible for generating the mechanical power, while the feedback device provides information about the position and speed of the load. The controller is responsible for processing the feedback signals and adjusting the motor's output accordingly to achieve the desired position or speed.

One of the main advantages of a servo motor is its precision and accuracy. The feedback system allows the motor to maintain a precise position, even under varying load conditions. This makes it ideal for applications such as CNC machines, robotics, and aerospace. It also has high torque to weight ratio so, it produces a lot of torque relative to their size and weight, making them suitable for applications where space and weight are limited. There are different types of servo motors available in the market, each with its unique features and specifications and the type of servo motors used are the SG90 and MG945.

4.3.1 SG90 Servo Motor

The SG90 servo motor is a small, low cost servo motor commonly used in various hobbyist projects. The image of the SG90 servo motor is given in the figure 4.3. Despite its small size, it is a versatile motor that is used for a wide range of applications, from robotics and automation to remote controlled vehicles and drones. It has a plastic gear train and produce’s up to 1.8 kg/cm of torque. It operates at a voltage range of 4.8 to 6 volts and has a maximum rotation angle of 180 degrees. It also has a positional accuracy of +/- 3 degrees, making it suitable for applications that require precise and repeatable movements

Figure 4.3 SG90 Servo motor

One of the main advantages of the SG90 servo motor is its low cost. It is one of the most affordable servo motors available in the market, making it an excellent choice. Another advantage of the SG90 servo motor is its small size and lightweight. It measures only 23mm x 12.2mm x 29mm and weighs only 9 grams, making it ideal for applications where space and weight are limited. However, the SG90 servo motor also has some limitations. Its plastic gear train is not as durable as metal gear trains, which limit's its lifespan in applications where it is subjected to high loads or frequent use.

4.3.2 MG945 Servo Motor

The MG945 servo motor is a powerful and high torque servo motor that is commonly used in industrial and commercial applications. The image of the MG945 servo motor is given in the figure 4.4. It is a versatile motor that handle's heavy loads and provide precise and accurate movements. It has a metal gear train and produce's up to 13 kg/cm of torque. It operates at a voltage range of 4.8 to 7.2 volts and has a maximum rotation angle of 180 degrees. It also has a positional accuracy of +/- 0.3 degrees, making it suitable for applications that require high precision and accuracy.

Figure 4.4: MG945 Servo motor

One of the main advantages of the MG945 servo motor is its high torque output. It handle’s heavy loads and provide consistent and reliable performance in applications where high torque is required, such as robotics, automation, and manufacturing. Another advantage of the MG945 servo motor is its metal gear train, which provides greater durability and longevity compared to plastic gear trains. This makes it suitable for applications where it is subjected to high loads or frequent use. However, the MG945 servo motor also has some limitations. It is relatively large and heavy compared to other servo motors, which limit’s its use in applications where space and weight are limited. It is also more expensive compared to other lower torque and smaller servo motors.

4.4 ROBOTIC ARM

A robotic arm is a mechanical arm like device that is programmed to move in a specific way and perform specific functions. Robotic arms are found in various industries, from manufacturing to healthcare. In manufacturing, robotic arms are used to assemble products, weld materials, and transport goods and in the medical field it is used to perform minimally invasive surgeries and assist patients with physical therapy. One of the key advantages of robotic arms is their precision. They are used perform tasks with a high degree of accuracy, which is especially important in industries such as aerospace and automotive, where even the smallest error has significant consequences. A Three Dimensional (3D) printed robotic arm with 6 DOF is used in the proposed system as it allows the production of customized and complex designs that would have been difficult or impossible to manufacture using traditional methods. The 6 DOF, allows the arm to perform a wide range of tasks. The figure 4.5 shows the image of the 3D model that was used to print the robotic arm.

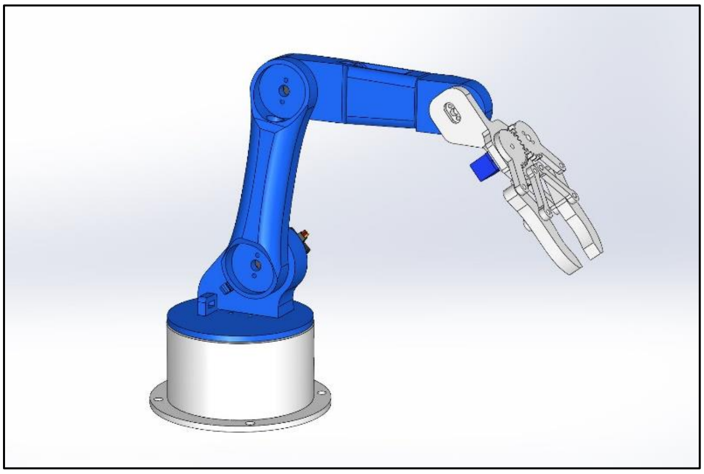


Figure 4.5 3D Model of the robotic arm

The arm is programmed to perform specific tasks or movements, and its flexibility allows it to perform tasks that are difficult for humans to accomplish, such as reaching into tight spaces or manipulating objects with precision. One of the key advantages of a 3D printed robotic arm is its affordability. Traditional robotic arms are expensive to manufacture and maintain, but a 3D printed arm is produced at a lower cost using readily available materials.

Additionally, the modular design of a 3D printed arm means that individual components are easily replaced or upgraded, extending the lifespan of the device. Another advantage of a 3D printed robotic arm is its adaptability. The arm are designed to fit specific use cases, which makes it more efficient and effective than a generic robotic arm.

4.5 SUMMARY

This chapter provided the detailed explanation of the components used in the hardware setup of the facial movement based robotic arm control using Python and Arduino. It also provided the detailed description of the specification and the function of each of the component.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 INTRODUCTION

This chapter reviews the detailed description of the various software and the programs used in the Facial movement based robotic arm control using Arduino and Python. This chapter also provides the details of the various parts of the program / source code and the mathematical equations used for various calculations.

5.2 ARDUINO IDE

Arduino IDE (Integrated Development Environment) is a software platform used for developing, programming, and uploading code to Arduino microcontrollers. Arduino IDE is an open source platform and is available for free download on the Arduino website. It runs on multiple operating systems such as Windows, macOS, and Linux. The IDE provides a simple and intuitive interface for creating and uploading code to Arduino boards. It includes a text editor, a compiler, and a serial monitor. The text editor provides a code editor with syntax highlighting, auto completion, and error highlighting.

The compiler converts the code into machine readable code that is executed by the Arduino board. The serial monitor provides a way to communicate with the board and see output and debug messages. Arduino IDE supports a variety of programming languages, including C++, C, and a simplified version of C++ called "Arduino Programming Language" which is similar to C++. The IDE also has a vast library of pre built functions and examples that are used to program the Arduino board.

5.3 PYTHON

Python is a high level programming language that has gained immense popularity over the years. Developed by Guido van Rossum in 1991, Python is an interpreted language that emphasizes code readability, simplicity, and ease of use. It has become the go to language for many developers, particularly in fields such as web development, scientific computing, data analysis, and artificial intelligence. One of the key strengths of Python is its simple and easy to understand syntax. It uses whitespace and indentation to structure code, which makes it very readable and easy to maintain. Additionally, Python has a vast standard library that includes modules for a wide range of tasks such as network programming, file I/O, regular expressions, and more. Another advantage of Python is its cross platform compatibility.

Python code run's on various operating systems such as Windows, Mac OS, and Linux, without needing to be modified. This makes it a flexible and convenient choice for developers who work across different platforms. Python's popularity has also led to a vast ecosystem of third party libraries and tools. These libraries include powerful frameworks such as Django and flask for web development, NumPy and Pandas for scientific computing, and TensorFlow and PyTorch for machine learning and artificial intelligence.

5.4 PYFIRMATA

Pyfirmata is a Python library that enables communication with an Arduino board using the firmata protocol. The firmata protocol is a standard protocol used to communicate with microcontrollers like Arduino using a simple serial interface. Pyfirmata simplifies the process of programming an Arduino by allowing developers to use Python code to interact with the board.

It provides an object oriented interface to the firmata protocol, making it easy to set up and configure an Arduino board. It provides a range of features, including digital input and output, analog input and output, PWM output, and more. It also allows developers to create custom messages and send them to the board. One of the key benefits of pyfirmata is its simplicity. It abstracts away many of the complexities of working directly with the firmata protocol, allowing developers to focus on their application logic rather than the details of the communication protocol.

It also makes it easy to integrate an Arduino board with other Python libraries and frameworks. pyfirmata is an open source library, and its source code is freely available on GitHub. It is actively maintained by a community of developers and receives regular updates and bug fixes. Additionally, it has extensive documentation, including a quick start guide, tutorials, and API documentation, making it easy for developers to get started with pyfirmata. In summary, it is a powerful and easy to use library for programming Arduino boards using python. Its simplicity and extensive documentation make it an ideal choice for developers who want to work with microcontrollers without the need for extensive knowledge of low level communication protocols.

5.5 DLIB

Dlib is a popular open source python library used for machine learning, computer vision, and image processing applications. It was created by Davis King and is known for its high quality implementation of a wide range of algorithms and its easy to use API. One of the most notable features of Dlib is its support for a range of machine learning algorithms. It includes implementations of popular algorithms such as k nearest neighbours, decision trees, support vector machines, and deep neural networks.

This makes it a powerful tool for a wide range of applications, from facial recognition to object detection and classification. Another key feature of Dlib is its support for computer vision and image processing. It includes a range of algorithms for feature extraction, image segmentation, and object tracking. These features make it a popular choice for developing applications that require advanced image analysis.

Dlib also includes a number of tools for optimizing the performance of machine learning models for example, it includes tools for cross validation, parameter tuning, and model selection. These tools make it easier to develop high quality machine learning models, even for developers with limited experience in this area.

One of the benefits of using Dlib is its ease of use. It has a well designed API that is easy to understand and use. Additionally, it includes extensive documentation, tutorials, and sample code, which makes it easy for developers to get started with the library. Since it is an open source library and is freely available on GitHub. It has an active community of developers who contribute to the library and provide support to other users. Additionally, Dlib has been used in a wide range of commercial and open source projects, which is a testament to its power and reliability.

5.6 OPENCV

OpenCV is an open source computer vision and machine learning library that was originally developed by Intel in 1999. It is written in C++ and has bindings for several other programming languages, including Python, Java, and MATLAB. It provides a wide range of algorithms for image processing, object detection, face recognition, and more. One of the main benefits of OpenCV is its versatility.

It provides a wide range of algorithms for image processing and computer vision, making it a useful tool for a variety of applications, including robotics, surveillance, medical imaging, and more. Some of the key algorithms included in OpenCV are feature detection and matching, image segmentation, object tracking, and face detection. Another benefit of OpenCV is its ease of use.

Additionally, it includes extensive documentation, tutorials, and sample code, which makes it easy for developers to get started with the library. OpenCV also provides a GUI interface that allows users to interact with the library visually and test different algorithms and it is also highly optimized for performance. It includes support for multi core processing, which significantly speed up processing times. Additionally, it take’s advantage of hardware acceleration provided by graphics processing units (GPUs), which further increases performance for certain algorithms. Since it is an open source library and is freely available on GitHub. It has an active community of developers who contribute to the library and provide support to other users.

5.7 FACE DETECTION

The Viola Jones algorithm has been used for detecting the face in a live video. It uses a complex combination of Haar features for object detection and Dlib for facial landmark. Haar features are a type of feature descriptor used in computer vision and image processing to detect objects or patterns within an image. It has a set of rectangular features that are computed at various positions and scales over an image. The features are computed by summing the pixel intensities within the white and black rectangular regions of the feature and subtracting the sum of the pixel intensities within the grey regions.

To detect an object or pattern in an image using Haar features, a cascade classifier is trained using a set of positive and negative examples. It is trained to distinguish between the positive and negative examples by sequentially applying a series of weak classifiers, based on different Haar feature. The output of each weak classifier is combined to obtain a strong classifier that accurately detects the object or pattern. The cascade classifier works by applying the weak classifiers in a cascade, where each subsequent classifier is applied only if the previous classifier passed. This results in a highly efficient detection system that quickly rejects non object regions of the image and focus only on the regions that are likely to contain the object or pattern. The technique relies upon placing a sub frame of 24x24 pixels within an image, and afterwards storing rectangular features inside it in every position with every size possible. These features consist of two, three or four rectangles. Some examples of such features are shown below in figure 5.1.

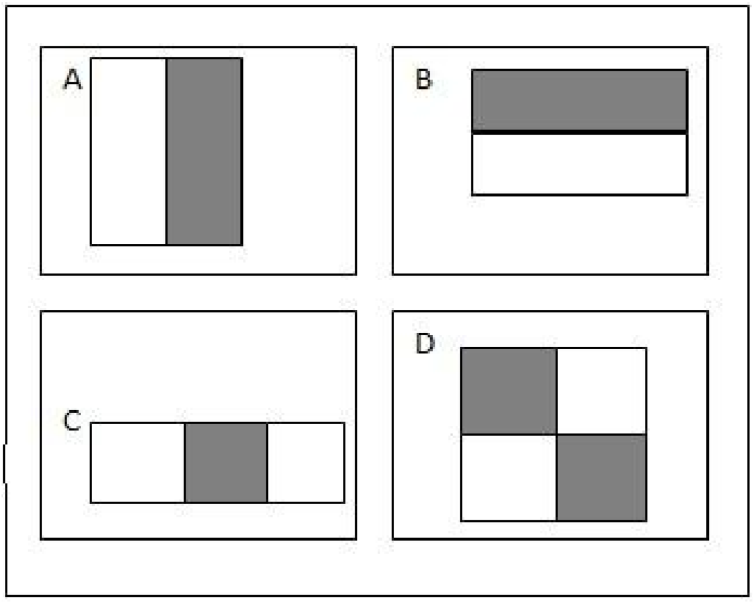


Figure 5.1 Haar features

Figure 5.1 shows two rectangle features, three rectangle features, and four rectangle feature.

The idea behind this is to not only observe the colour or luminance but the difference between the highlighted rectangles. Feature type A cannot be defined as a 1x1 pixel block; it must at least 1x2 pixels. Here, type D must be at least 2x2 pixels, and this rule holds accordingly other features holds this rule. Feature type A cannot be defined as 1x3 pixel block as partition is not possible in middle pixel.

Also, the feature type C width must be divisible by 3, and other features hold this rule. Feature with a width and/or height of 0 cannot be defined. Therefore, iteration is required for x and y to 24 minus the size of the feature. The Dlib facial landmark detector marks the face into 68 points as shown in figure 5.2. These points will be used to measure the EAR and MAR. Points from 37 to 42 mark the right eye and 43 to 47 mark the left eye. The points from 49 to 60 mark the outer portion of the lips and 61 to 68 mark the inner lips. The face is actively tracked in real time.

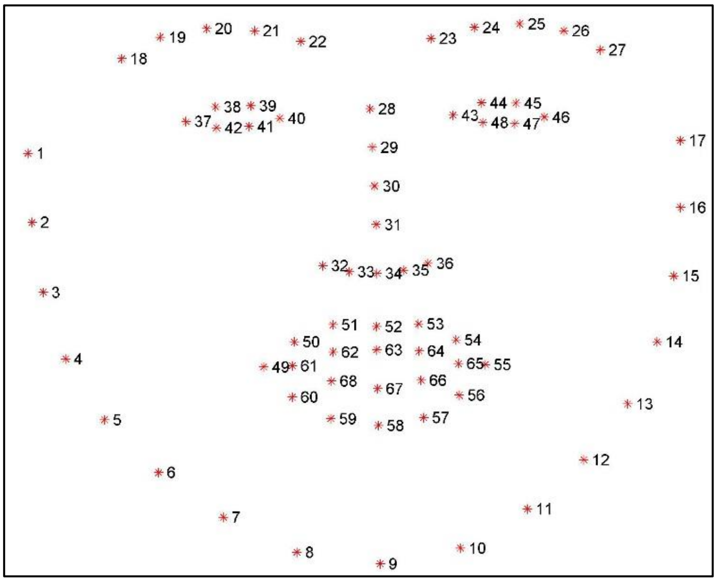


Figure 5.2 Visualizing the 68 facial landmarks

5.8 EYE DETECTION

The first step of detection of the gaze is to locate the eye regions, in order to extract the necessary features. The fundamental requirement of an iris tracking and gaze detection system is to accurately detect the eye sockets, which is easily achieved by Haar like object detectors. It allows a classifier trained with sample views of a particular object to be detected in a whole image. An eye image is captured with a zoom in camera of high resolution. This provides iris images with a greater number of pixels.

5.9 CENTER OF IRIS DETECTION

Hough circle algorithm which detects all the circles in the image and draws them in the accumulator variable. It was experimentally noted in the experiments that the circle with largest radius detected in the frame matched the boundary of the pupil. These circles were ignored by setting a threshold size of the circle radius which helped in filtering out smaller circles from processing. The accumulator variable stores the higher numerical value in the array and returns the coordinates of the cornea in the frames of the video. The next step in the process is to determine the current live location of the centre detected with respect them to centre point of the frame and find EAR then the MAR.

$$Y - Y_1 = \frac{(Y_2 - Y_1)/(X_2 - X_1)}{(X - X_1)} \quad \text{............ } \textbf{Equation (5.1)}$$

The equation 5.1 is used to initiate the arm movement from current point in the direction where the user gazes.

We assume an $(N\ /\ N)$ square grid that is used to obtain the equation of the line by using the coordinates at the center of the grid (X_0, Y_0) and current detected coordinate of the center of the eye (X_d, Y_d) and substituting in the equation. Initially the coordinates (X_0, Y_0) are kept as $(N/2\ , N/2)$ and the values $Y_0, Y_1, Y_2\ \ldots\ldots\ Y_N$ are calculated by substituting values for $X_0, X_1, X_2 \ldots\ldots X_N$ into the line equation.

5.10 EYE ASPECT RATIO

The EAR is a measure used in facial recognition and computer vision to detect eye blinks and track eye movements. It is calculated as the ratio of the distance between the vertical landmarks of the eye to the distance between the horizontal landmarks of the eye. More specifically, the EAR is calculated by first identifying the six landmarks of the eye: the top and bottom of the upper eyelid, the top and bottom of the lower eyelid, and the inner and outer corners of the eye. Then, the Euclidean distances between the vertical landmarks and the horizontal landmarks are calculated. The EAR is computed as the ratio of the average of the vertical distances to the average of the horizontal distances. The aspect ratio is generally constant when the eye is open and starts tending to zero while closing of eye. By using the equation 5.2. six points that are placed over the eye as shown in the figure 3.5. The numerator of the equation 5.2 computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks.

$$EAR = \frac{|\mathrm{p2}-\mathrm{p1}|+|\mathrm{p3}-\mathrm{p5}|}{2|p1-p4|} \quad \text{………..……} \quad \textbf{Equation (5.2)}$$

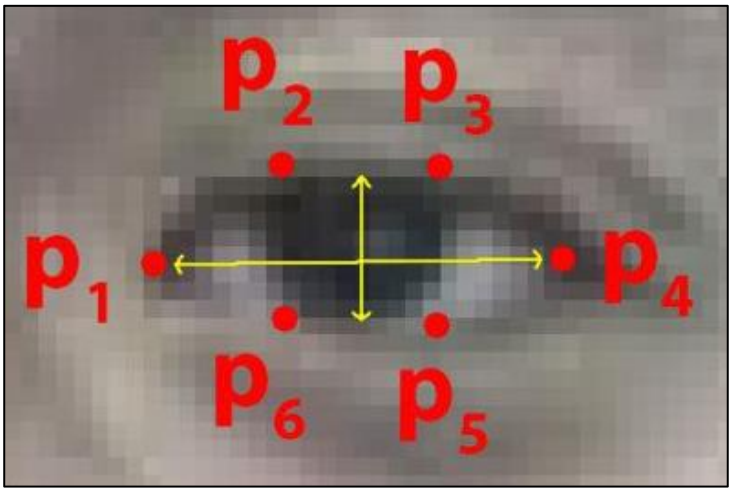


Figure 5.3 Landmarks points placed over the eyes

By using this method image processing techniques are avoided and simply rely on the ratio of eye landmark distances to determine if a person's eye is closed or open or if the user is blinking.

5.11 MOUTH ASPECT RATIO

MAR is used to detect whether a person is opening their mouth or not. This is a similar mathematical approach as that of EAR as places points on the user's mouth as shown in figure 5.4 and also measures the ratio of the length of the mouth to the width of the mouth.

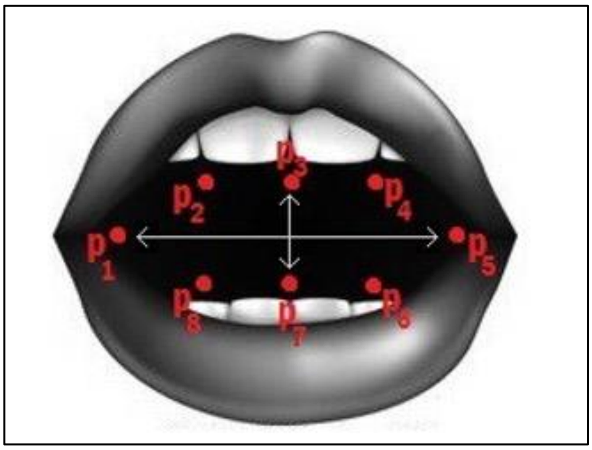


Figure 5.4 Landmark points placed over the mouth

It calculates the length of the mouth by considering average distance between points by using the facial points obtained from the face detection phase. It is known that points from 49 to 68 represent the mouth coordinates. So, eight points P1 to P8 is placed on the points 61 to 68. After the points are placed the MAR is calculated by using the equation 5.3. All the points placed over the mouth are considered for better accuracy and the distance between the points P1 to P8 are calculated continuously because when distance between the points increases the MAR also increases and this change in value is used to perform various actions.

$$MAR = \frac{|\mathrm{p2} - \mathrm{p8}| + |\mathrm{p3} - \mathrm{p7}| + |\mathrm{p4} - \mathrm{p6}|}{3|p1 - p5|} \quad \text{……….} \ \textbf{Equation (5.3)}$$

5.12 BLINK CLASSIFIER

The blink classifier is a part of the program that is responsible for performing the function of selecting the part the robotic arm that the user wants to control. Since, the average time it takes for a complete human blink is about 300 to 400 milliseconds or 3/10ths to 4/10ths of a second. Of course, this is an average only and differ from person to person. So, to perform selection operations, the user must blink with one eye for at least 700 to 800 milliseconds. In that period viola jones algorithm will detect only one eye.

If the left eye is only detected, then it means that the right eye is closed so the program selects the base / shoulder of the robotic arm and if right eye is detected, then it means that the left eye is closed so the program selects the elbow of the robotic arm.

If both eyes are detected in a squinting position the program selects the wrist of the robotic arm and if both eyes are not detected, then it will consider that it is normal blink. Table 5.1 shows the selection of the part of the robotic arm performed for each eye movement.

Table 5.1 Selection conditions

Right Eye	**Left Eye**	**Selection**
False	True	Shoulder / Base
True	False	Elbow
True	True	Wrist
False	False	Blink

5.13 SUMMARY

This chapter provided the detailed explanation of the various programs and the mathematical equations used in the software setup of the facial movement based robotic arm control using python and Arduino. It also provided the selection conditions that are performed by the blinking the eyes for selecting the part of the robotic arm to be controlled.

CHAPTER 6

PERFORMANCE ANALYSIS

6.1 INTRODUCTION

This chapter consists of the description of the hardware components used and also the detailed explanation of the execution process, inputs, and the outputs of the proposed method.

6.2 HARDWARE COMPONENTS

The table 6.1 shows the various components that are used in the system and its specification.

Table 6.1 Hardware components and it's specification

S.NO	Component	Specification
1	Robotic arm	6 Degrees of freedom
2	Arduino	UNO
3	SG90 (Servo motor)	180° – Low torque
4	MG945 (Servo motor)	180° – High torque
5	Jumper cables	6 inch – male to male
6	Web camera	Zoom lens – 720p

6.3 RUNNING THE PROGRAM

The program is run using the Python IDE as it uses universal open source libraries, but all the libraries must be downloaded prior to running of the program to avoid any errors. The figure 6.1 shows the running of the program.

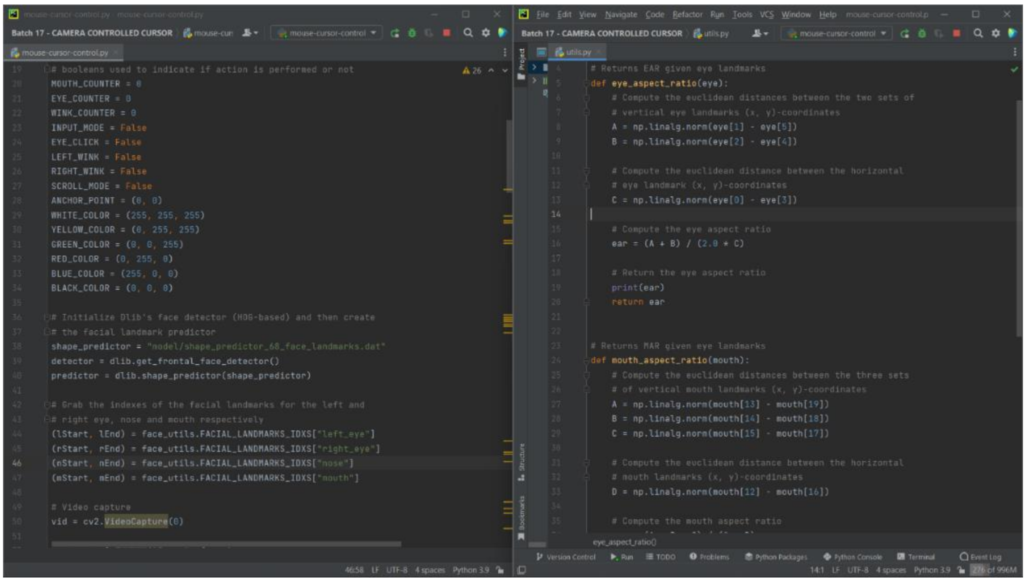


Figure 6.1 Running of the program

6.4 READING THE INPUT

The program displays the user's image with the tracking points, and the program is started by opening the mouth and inputs are provided for moving the robotic arm by the movement of head. The figure 6.2 shows the program reading the user's input

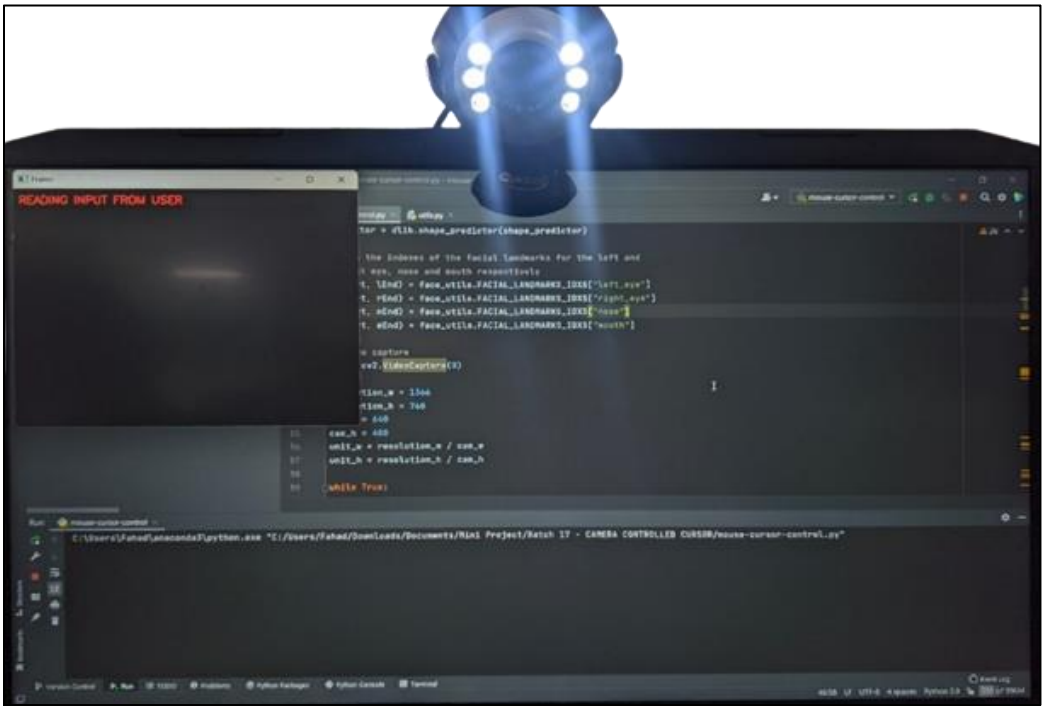


Figure 6.2 Reading the input from user

6.5 MOVEMENT OF ROBOTIC ARM

The table 6.1 shows the user’s actions and its corresponding function that moves the robotic arm. By taking the inputs from the user, different movements are connected to various functions that perform the action of moving the robotic arm. The figure 6.3 shows the robotic arm

Table 6.2 Facial movements and it’s corresponding function

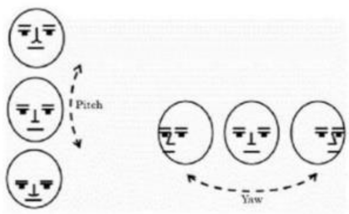
Action	Function
 Opening Mouth	Start/Stop program
 Right Eye Wink	Selection of elbow
 Left Eye Wink	Selection of Wrist
 Squinting Eyes	Selection of Shoulder / base
 Head Movements (Pitch and Yaw)	Movement of selected part of robotic arm

Figure 6.3 Robotic arm

6.4 SUMMARY

In this chapter the step by step work progress right from components and its specification as shown in the table 6.1 to the working of the system were discussed completely. This chapter also shows the image of the hardware setup of the robotic arm as exposed in figure 6.3 and the facial movements and its corresponding action on the robotic arm movement is also given in the table 6.2. The next chapter discuss about the conclusion of the project discussed.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

This paper presents an eye gaze controlled robotic arm to help in the rehabilitation process of users with SSMI. With the use of a camera and python technology, the system architecture is prepared. User is able to view head and eye movements through the camera display, the user is able to select a part of the robotic arm and move it as needed and perform various actions without any calibration. This application benefit disabled individuals who wants to achieve greater mobility or use it in the same way as ordinary people. The Goal and soul purpose of making this project is to simplify the use of the robotic arm with only one software and also make it cost effective so it will be used by people with severe disabilities for getting more job opportunities in industries such as in automotive industry in which robotic arm is used for assembly of various parts and it is used in the medical field to perform non invasive surgeries and laboratory work.

7.2 FUTURE SCOPE

Since the type of gripper used is limited to jaw type gripper, It may be vastly improved by providing a modular gripper system that enables the user to select any type of gripper such as vacuum gripper, magnetic gripper or any other gripper that is necessary for a specific application. Improvement in the system’s accuracy and precision will help in use various high precision tasks in industries such as fabric painting, welding and non invasive surgeries.

REFERENCES

[1] V. Krishna Sharma, K. Saluja, V. Mollyn and P. Biswas (2021), "Eye gaze controlled robotic arm for persons with severe speech and motor impairment", ACM Symposium on Eye Tracking Research and Applications ser. ETRA'20 Full Papers, 2021. (pp 113 – 119)

[2] Jeevithashree, D. V., Saluja, K. S., Biswas, P. (2019). A case study of developing gaze controlled interface for users with severe speech and motor impairment. Technology and Disability, (pp 63-76).

[3] Rosebrock A. (2019), Facial landmarks with dlib, OpenCV, and Python, Available at https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[4] Baltrusaitis, T., Zadeh, A., Lim, Y. C., Morency, L. P. (2018). Facial behavior analysis. In 2018 13th IEEE International Conference on Automatic Face andGesture Recognition (FG 2018) (pp. 59-66). IEEE.

[5] Biswas P. and Jeevithashree DV (2018), Eye Gaze Controlled MFD for Military Aviation, ACM International Conference on Intelligent User Interfaces (IUI) 2018 (pp 165 – 170)

[6] Koch, P. J., van Amstel, M. K., Dębska, P., Thormann, M. A., Tetzlaff, A. J., Bøgh, S., Chrysostomou, D. (2017). A skill-based robot co-worker for industrial maintenance tasks. Procedia Manufacturing, 11, (pp 83-90).

[7] Alsharif S., Kuzmicheva O. and Gräser A. (2016), Gaze Gesture-Based Human Robot Interface, Zweite transdisziplinäre Konferenz,Technische Unterstützungssysteme, die Menschen wirklich wollen, 2016 (pp 34–40)

[8] Bremner, P., Celiktutan, O., Gunes, H. (2016). Personality perception of robot avatar teleoperators. In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (pp. 141-148). IEEE

[9] Dziemian, Sabine, William W. Abbott, and A. Aldo Faisal (2016), Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing & drawing. 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob). (pp 102 – 107).

[10] Khonglah, J. R., & Khosla, A. (2015). A low cost webcam based eye tracker for communicating through the eyes of young children with ASD. In Next Generation Computing Technologies (NGCT), 2015 1st IEEE International Conference (pp 925-928)

[11] Hansen, J. P., Alapetite, A., MacKenzie, I. S., Møllenbach, E. (2014). The use of gaze to control drones. In Proceedings of the Symposium on Eye Tracking Research and Applications (pp. 27-34)

[12] Dostal, J., Kristensson, P. O., Quigley, A. (2013). Subtle gaze-dependent techniques for visualising display changes in multi-display environments. In Proceedings of the 2013 international conference on Intelligent user interfaces (pp. 137-148).

[13] Bannat, A., Gast, J., Rehrl, T., Rösel, W., Rigoll, G., Wallhoff, F. (2011). A multimodal human-robot-interaction scenario: Working together with an industrial robot. In International Conference on Human-Computer Interaction (pp. 303-311). Springer, Berlin, Heidelberg.

[14] San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., Hansen, J. P. (2010). Evaluation of a low-cost open-source gaze tracker. In Proceedings of the 2010 Symposium on Eye-Tracking Research (pp 77-80).

[15] Betke, Margrit, James Gips, and Peter Fleming (2009) "The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities." IEEE Transactions on neural systems and Rehabilitation Engineering 10.1: (pp 1-10).

[16] Bonneau, E., Taha, F., Gravez, P., Lamy, S. (2009). Surgicobot: Surgical gesture assistance cobot for maxillo-facial interventions. In Perspective in Image-Guided Surgery(pp. 353-360).

[17] Chen Y. and Newman W. S. (2008), A Human-Robot Interface Based on Electrooculography, Proceedings of the IEEE Intl Conf on Robotics and Automation. (pp 212 – 216)

[18] Dautenhahn, K., Woods, S., Kaouri, C., Walters, M. L., Koay, K. L., Werry, I. (2005). What is a robot companion-friend, assistant or butler?. In 2005 IEEE/RSJ international conference on intelligent robots and systems (pp. 1192-1197). IEEE.

[19] Kim, D. H., Kim, J. H., Yoo, D. H., Lee, Y. J., Chung, M. J. (2001). A human-robot interface using eye-gaze tracking system for people with motor disabilities. Transaction on Control, Automation and Systems Engineering,(pp 229-235).

[20] Fitts P.M. (2001), The Information Capacity of the Human Motor System In Controlling The Amplitude of Movement, Journal of Experimental Psychology 47 (1954): (pp 381-391).

APPENDICES

SOURCE CODE

```
from imutils import face_utils

from utils import *

import numpy as np

import imutils

import dlib

import cv2

from aur import *

MOUTH_AR_THRESH = 0.06

MOUTH_AR_CONSECUTIVE_FRAMES = 15

EYE_AR_THRESH = 0.22

EYE_AR_CONSECUTIVE_FRAMES = 12

WINK_AR_DIFF_THRESH = 0.02

WINK_AR_CLOSE_THRESH = 0.14

WINK_CONSECUTIVE_FRAMES = 10

MOUTH_COUNTER = 0

EYE_COUNTER = 0

WINK_COUNTER = 0

INPUT_MODE = False
```

```
EYE_CLICK = False

LEFT_WINK = False

RIGHT_WINK = False

SCROLL_MODE = False

ANCHOR_POINT = (0, 0)

WHITE_COLOR = (255, 255, 255)

YELLOW_COLOR = (0, 255, 255)

GREEN_COLOR = (0, 0, 255)

RED_COLOR = (0, 255, 0)

BLUE_COLOR = (255, 0, 0)

BLACK_COLOR = (0, 0, 0)

shape_predictor = "model/shape_predictor_68_face_landmarks.dat"

detector = dlib.get_frontal_face_detector()

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]

(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

(nStart, nEnd) = face_utils.FACIAL_LANDMARKS_IDXS["nose"]

(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]

vid = cv2.VideoCapture(0)

resolution_w = 1366

resolution_h = 768
```

```
cam_w = 640

cam_h = 480

unit_w = resolution_w / cam_w

unit_h = resolution_h / cam_h

while True:

    _, frame = vid.read()

    frame = cv2.flip(frame, 1)

    frame = imutils.resize(frame, width=cam_w, height=cam_h)

    rects = detector(gray, 0)

    if len(rects) > 0:

        rect = rects[0]

    else:

        cv2.imshow("Frame", frame)

        key = cv2.waitKey(20) & 0xFF

        continue

    shape = predictor(gray, rect)

    shape = face_utils.shape_to_np(shape)

    mouth = shape[mStart:mEnd]

    leftEye = shape[lStart:lEnd]

    rightEye = shape[rStart:rEnd]
```

```
    nose = shape[nStart:nEnd]
    temp = leftEye
    leftEye = rightEye
    rightEye = temp
    mar = mouth_aspect_ratio(mouth)
    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)
    ear = (leftEAR + rightEAR) / 2.0
    diff_ear = np.abs(leftEAR - rightEAR)
    nose_point = (nose[3, 0], nose[3, 1])
    mouthHull = cv2.convexHull(mouth)
    leftEyeHull = cv2.convexHull(leftEye)
    rightEyeHull = cv2.convexHull(rightEye)
    cv2.drawContours(frame, [mouthHull], -1, YELLOW_COLOR, 1)
    cv2.drawContours(frame, [leftEyeHull], -1, YELLOW_COLOR, 1)
    cv2.drawContours(frame, [rightEyeHull], -1, YELLOW_COLOR, 1)
    for (x, y) in np.concatenate((mouth, leftEye, rightEye), axis=0):
        cv2.circle(frame, (x, y), 2, GREEN_COLOR, -1)
if EYE_COUNTER > EYE_AR_CONSECUTIVE_FRAMES:
        shoulder = True
```

```
        elbow = False
        wrist = False
        EYE_COUNTER = 0
    else:
        EYE_COUNTER = 0
        WINK_COUNTER = 0
if INPUT_MODE:
    cv2.putText(frame, "READING INPUT!", (10, 30),
    w, h = 35, 35
    multiple = 1
    drag = 18
    if shoulder:
        if dir == 'right' and s1 > 0:
            rotateservo(9, s1 - 2)
            s1 = s1 - 2
            if s1 - 2 <= 0:
                rotateservo(9, 0)
        if dir == 'left' and s1 < 180:
            rotateservo(9, s1 + 2)
            s1 = s1 + 2
```

```
        if s1 +  >= 180:
            rotateservo(9, 180)
            s1 = 180
    if dir == 'up' and s3 < 180:
        rotateservo(3, s3 + 2)
        s3 = s3 + 2
        if s3 + 2 >= 180:
            rotateservo(3, 180)
            s3 = 180
    elif dir == 'down' and s3 > 0:
        rotateservo(3, s3 - 2)
        s3 = s3 - 2
        if s3 - 2 <= 0:
            rotateservo(3, 0)
            s3 = 0
if elbow:
    if dir == 'left' and s5 > 0:
        rotateservo(5, s5 - 2)
        s5 = s5 - 2
        if s5 - 2 <= 0:
```

```
            rotateservo(5, 0)
            s5 = 0
    if dir == 'right' and s5 < 180:
        rotateservo(5, s5 + 2)
        s5 = s5 + 2
        if s5 + 2 >= 180:
            rotateservo(5, 180)
            s5 = 180
    elif dir == 'down' and s4 < 180:
        rotateservo(4, s4 + 2)
        s4 = s4 + 2
        if s4 + 2 >= 180:
            rotateservo(4, 180)
            s4 = 180
    elif dir == 'up' and s4 > 0:
        rotateservo(4, s4 - 2)
        s4 = s4 - 2
        if s4 - 2 <= 0:
            rotateservo(4, 0)
            s4 = 0
```

```
        if wrist:
            if dir == 'right' and s7 > 0:
                rotateservo(7, s7 - 2)
                s7 = s7 - 2
                if s7 - 2 <= 0:
                    rotateservo(7, 0)
                    s7 = 0
            elif dir == 'up' and s6 < 180:
                rotateservo(6, s6 + 2)
                s6 = s6 + 2
                if s6 + 2 >= 180:
                    rotateservo(6, 180)
                    s6 = 180
            elif dir == 'down' and s6 > 0:
    if key == 27:
        break
cv2.destroyAllWindows()
vid.release()
```

LIST OF PUBLICATIONS

[1] Mohammed Al Fahad, Prem Prakash S, Kavinraj V, Deepa V has presented a paper titled "FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON" in the International Conference on Computing and Information Technology (ICCIT – 2023) organized by the Faculty of Computers and Information Technology, University of Tabuk, Saudi Arabia on March 28th 2023.

[2] Mohammed Al Fahad, Prem Prakash S, Kavinraj V, Deepa V has published a paper titled "FACIAL MOVEMENT BASED ROBOTIC ARM CONTROL USING ARDUINO AND PYTHON" in the International Journal on Innovative Research in Engineering (IJIRE), Volume no: 4, Issue no: 1, pp 151-154.

IEEE SAUDI ARABIA SECTION

ICCIT

معتمد ACCREDITED 2020 - 2026

جامعة تبوك University of Tabuk

Certificate

This is to certify that

Mohammed Al Fahad

has/have Presented a paper entitled

"Facial Movement Based Robotic Arm Control Using Arduino and Python"

in the 2023-International Conference on Computing and Information Technology (ICCIT-2023) which was organized by the Faculty of Computers and information Technology, University of Tabuk and Technically sponsored by the IEEE Saudi Section, from 27th to 29th March, 2023 in Tabuk, Kingdom of Saudi Arabia.

The Technical Program Chair

Dr. Saad M Almutairi Ph.D., MIEEE.,

Dean, IT Deanship,
University of Tabuk, Tabuk City, Saudi Arabia.

The Conference Chair

Dr. Majed Mohammed Aborokbah Ph.D., MIEEE.,

Dean, Faculty of Computers and Information Technology,
University of Tabuk, Tabuk City, Saudi Arabia.
Email: ic.chair@ut.edu.sa

The Organizing Chair

Dr. Manimurugan S Ph.D., MISTE., MIEEE.,

Associate professor, Faculty of Computers and Information Technology, University of Tabuk, Tabuk City, Saudi Arabia.

ICCIT IEEE SAUDI ARABIA SECTION

معتمد ACCREDITED 2020 - 2026

جامعة تبوك University of Tabuk

Certificate

This is to certify that

Prem Prakash S

has/have Presented a paper entitled

"Facial Movement Based Robotic Arm Control Using Arduino and Python"

in the 2023-International Conference on Computing and Information Technology (ICCIT-2023) which was organized by the Faculty of Computers and information Technology, University of Tabuk and Technically sponsored by the IEEE Saudi Section, from 27th to 29th March, 2023 in Tabuk, Kingdom of Saudi Arabia.

The Technical Program Chair

Dr. Saad M Almutairi Ph.D., MIEEE.,

Dean, IT Deanship,
University of Tabuk, Tabuk City, Saudi Arabia.

The Conference Chair

Dr. Majed Mohammed Aborokbah Ph.D., MIEEE.,

Dean, Faculty of Computers and Information Technology,
University of Tabuk, Tabuk City, Saudi Arabia.
Email: ic.chair@ut.edu.sa

The Organizing Chair

Dr. Manimurugan S Ph.D., MISTE., MIEEE.,

Associate professor, Faculty of Computers and Information Technology, University of Tabuk, Tabuk City, Saudi Arabia.

ICCIT IEEE SAUDI ARABIA SECTION

2020 - 2026 ACCREDITED معتمد

جامعة تبوك University of Tabuk 2006

Certificate

This is to certify that

Kavinraj V

has/have Presented a paper entitled

"Facial Movement Based Robotic Arm Control Using Arduino and Python"

in the 2023-International Conference on Computing and Information Technology (ICCIT-2023) which was organized by the Faculty of Computers and information Technology, University of Tabuk and Technically sponsored by the IEEE Saudi Section, from 27th to 29th March, 2023 in Tabuk, Kingdom of Saudi Arabia.

The Technical Program Chair

Dr. Saad M Almutairi Ph.D., MIEEE.,

Dean, IT Deanship,
University of Tabuk, Tabuk City, Saudi Arabia.

Faculty of Computing & Information Technology

The Conference Chair

Dr. Majed Mohammed Aborokbah Ph.D., MIEEE.,

Dean, Faculty of Computers and Information Technology,
University of Tabuk, Tabuk City, Saudi Arabia.
Email: ic.chair@ut.edu.sa

The Organizing Chair

Dr. Manimurugan S Ph.D., MISTE., MIEEE.,

Associate professor, Faculty of Computers and Information Technology, University of Tabuk, Tabuk City, Saudi Arabia.

ICCIT IEEE SAUDI ARABIA SECTION

ACCREDITED معتمد 2020 - 2026

جامعة تبوك University of Tabuk 2006

Certificate

This is to certify that

Deepa V

has/have Presented a paper entitled

"Facial Movement Based Robotic Arm Control Using Arduino and Python"

in the 2023-International Conference on Computing and Information Technology (ICCIT-2023) which was organized by the Faculty of Computers and information Technology, University of Tabuk and Technically sponsored by the IEEE Saudi Section, from 27th to 29th March, 2023 in Tabuk, Kingdom of Saudi Arabia.

The Technical Program Chair
Dr. Saad M Almutairi Ph.D., MIEEE.,
Dean, IT Deanship,
University of Tabuk, Tabuk City, Saudi Arabia.

Faculty of Computing & Information Technology

The Conference Chair
Dr. Majed Mohammed Aborokbah Ph.D., MIEEE.,
Dean, Faculty of Computers and Information Technology,
University of Tabuk, Tabuk City, Saudi Arabia.
Email: ic.chair@ut.edu.sa

The Organizing Chair
Dr. Manimurugan S Ph.D., MISTE., MIEEE.,
Associate professor, Faculty of Computers and Information Technology, University of Tabuk, Tabuk City, Saudi Arabia.

Certificate

OF PUBLICATION

THIS CERTIFICATE IS CONFIRM THAT

Mohammed Al Fahad

PUBLISHED FOLLOWING ARTICLE

Facial Movement Based Robotic Arm Control Using Arduino and Python

Volume 4, Issue 1 (January-February 2023), PP: 151-154.

A Peer Reviewed referred Journal

International Journal of Innovative Research in Engineering
ISSN No:2582-8746

Editor-in-chief/IJIRE

Certificate

OF PUBLICATION

THIS CERTIFICATE IS CONFIRM THAT

Prem Prakash S

PUBLISHED FOLLOWING ARTICLE

Facial Movement Based Robotic Arm Control Using Arduino and Python

Volume 4, Issue 1 (January-February 2023), PP: 151-154.

A Peer Reviewed referred Journal

**International Journal of
Innovative Research in Engineering
ISSN No:2582-8746**

Editor-in-chief/IJIRE

Certificate

OF PUBLICATION

THIS CERTIFICATE IS CONFIRM THAT

Kavinraj V

PUBLISHED FOLLOWING ARTICLE

Facial Movement Based Robotic Arm Control Using Arduino and Python

Volume 4, Issue 1 (January-February 2023), PP: 151-154.

A Peer Reviewed referred Journal

International Journal of
Innovative Research in Engineering
ISSN No:2582-8746

Editor-in-chief/IJIRE

Certificate

OF PUBLICATION

THIS CERTIFICATE IS CONFIRM THAT

Deepa V

PUBLISHED FOLLOWING ARTICLE

Facial Movement Based Robotic Arm Control Using Arduino and Python

Volume 4, Issue 1 (January-February 2023), PP: 151-154.

A Peer Reviewed referred Journal

International Journal of
Innovative Research in Engineering
ISSN No:2582-8746

Editor-in-chief/IJIRE