

HAND GESTURES BASED ROBOTIC ARM USING MEDIA PIPE AND OPEN CV

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Electronics and Communication Engineering

by

Grandhi Revanth(40130071)

K. V V PHANI KUMAR(40130109)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI, SALAI, CHENNAI - 600119

APRIL-2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

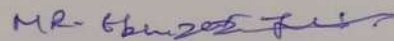
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

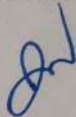
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Katta Veera Venkata Phani Kumar (40130109)** and **Grandhi Revanth(40130071)** who carried out the Project entitled "**HAND GESTURES BASED ROBOTIC ARM USING MEDIA PIPE AND OPEN CV**" under the guidance of **Dr.M.R.Ebenezar,M.E.,Ph.D.** From Nov 2023 to April 2024.



Internal Guide

Dr.M.R.EBENEZAR JEBARANI M.E.,Ph.D.,

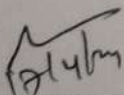


Head of the Department

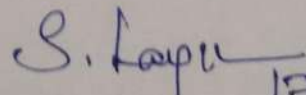
Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on

17/04/2024



Internal Examiner



External Examiner

DECLARATION

We are **Katta Veera Venkata Phani Kumar** (40130109) and **Grandhi Revanth** (40130071) here by declare that the Project Report entitled "**HAND GESTURES BASED ROBOTIC ARM USING MEDIA PIPE AND OPEN CV**" under the guidance of **Dr.M.R.EBENEZAR JEBARANI,M.E.,Ph.D.** is submitted in partial fulfillment of the requirements for the Bachelor of Engineering degree in Electronics and Communication.

DATE: 17/04/24

PLACE:

SIGNATURE OF THE CANDIDATES

1. G.Revanth
2. K.V.Phanikumar

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE ND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E.,Ph.D., Professor & Dean, School of Electrical and Electronics** and **Dr. T. RAVI, Ph.D., Professor & Head, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our **INTERNAL GUIDE Dr.M.R.EBENEZAR JEBARANI,M.E.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways.For the completion of the project.

ABSTRACT

The open cv robotic arm system is essentially a combination of hardware and software that can perform various tasks, such as picking up and moving objects, or tracking the motion of an object. Open CV(Open Source Computer Vision Library) is an open-source computer vision and image processing library that provides a range of tools for object detection, tracking, and analysis .The robotic arm is typically composed of several joints, each with its own range of motion, and an end-effector, which is the part of the arm that interacts with the environment. The arm is controlled by a microcontroller that receives commands from the computer vision system .To detect and track objects, the computer vision system analyzes the input images from a camera and uses various algorithms to identify relevant features, such as edges or colour gradients. These features are the used to estimate the position and orientation of the object in the image, which is translated into a set of commands for the robotic arm to move. Overall, The Open cv based robotic arm system has wide range of potential applicatons,from manufacturing and assembly lines to ware house automation and even medical procedures.

CONTENTS

CHAPTER No	TITLE	PAGE No
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 PROBLEM STATEMENT	1
	1.3 LITERATURE REVIEW	2
2	HARDWARE AND SOFTWARE REQUIREMENTS	3
	2.1 ROBOTIC ARM	3
	2.2 OPEN CV	3
	2.3 IMAGE PROCESSING	5
	2.4 ARDUINO	12
3	METHODOLOGY	19
4	ADVANTAGES & DISADVANTAGES	23
5	RESULT	24
6	CONCLUSION AND FUTURE SCOPE	26
	REFERENCES	27

LIST OF FIGURES

FIGURE No	FIGURE NAME	PAGE No
2.1	IMAGE PROCESSING	5
2.2	ARDUINO	13
2.3	HAND GESTURES	19
5.1	CONNECTIONS	24
5.2	ROBOTIC ARM	24

LIST OF TABLES

TABLE No	TABLE NAME	PAGE No
2.1	TECHNICAL SPECIFICATIONS	14

CHAPTER-1

INTRODUCTION

1.1 OBJECTIVE

The objective of an Open CV based robotic arm is to perform tasks that require vision-based guidance, such as object detection, recognition, and tracking. By combining a robotic arm with Open CV, it is possible to automate various industrial and manufacturing processes that involve the manipulation of objects. The primary objective of an Open CV based robotic arm is to increase efficiency and accuracy in manufacturing and industrial processes. By using computer vision algorithms, the robotic arm can perform tasks that are difficult or impossible for a human operator to perform with the same level of precision and consistency. This can lead to significant improvements in product quality, increased through put, and reduced costs. Another objective of an Open CV based robotic arm is to improve safety in the workplace. By automating dangerous or repetitive tasks, the risk of injury to human operators can be reduced.

1.2 Problem statement

Repetitive and Dangerous Tasks: In some industrial processes, there are tasks that are repetitive or dangerous for human operators to perform. These tasks can include tasks such as lifting heavy objects, working in hazardous environments, or performing tasks that require high precision. An OpenCV based robotic arm can address this problem by automating these tasks, thereby reducing the risk of injury to human operators and improving productivity. In manufacturing and assembly processes, there is a need to perform tasks such as picking and placing objects, assembling components, and performing quality inspections. An OpenCV based robotic arm can address this problem by using computer vision algorithms to perform these tasks with high precision and consistency

1.3 Literature review

OpenCV (Open Source Computer Vision) is a widely used open-source computer vision library that provides a wide range of algorithms for image processing, object detection, recognition, and tracking. Robotic arm control using OpenCV has become an active research area in recent years, and there have been several advancements in this field. OpenCV can be used in various aspects of robotic arm control, including object recognition, tracking, and manipulation. The following are some of the key areas where OpenCV is used in robotic arm control. Object recognition is a critical component of robotic arm control, and OpenCV provides a variety of algorithms for object detection and recognition. These algorithms use techniques such as edge detection, feature extraction, and machine learning to recognize objects in an image or video stream. Once an object is detected, the robotic arm can be programmed to perform a specific action on the object. Tracking involves following the movement of an object in real-time. OpenCV provides a range of algorithms for object tracking, such as the Lucas-Kanade algorithm and the mean - shift algorithm. These algorithms use various techniques to track objects, such as optical flow, color histograms, and template matching. Once an object is tracked, the robotic arm can be programmed to follow the movement of the object. HumanRobot Interaction OpenCV can also be used to enable human-robot interaction, allowing the robotic arm to detect and respond to human actions. For example, OpenCV can be used to recognize hand gestures, which can then be used to control the movement of the robotic arm.

CHAPTER-2

HARDWARE AND SOFTWARE DESCRIPTION

2.1 ROBOTIC ARM

A robotic arm is a mechanical arm that is designed to perform a variety of tasks that require precision, accuracy, and repeatability. It is typically made up of several rigid segments or links that are connected by joints, allowing the arm to move and rotate in various directions. At the end of the arm, there is usually an end-effector, which is the part of the arm that interacts with the environment. The end-effector can take many different forms, depending on the task it is designed to perform. For example, it could be a gripper for picking up and manipulating objects, a drill for drilling holes, or a welding tool for joining materials together. Robotic arms are used in a wide range of applications, including manufacturing, assembly, material handling, and surgery. They are often used in situations where it is difficult or dangerous for humans to perform the task, such as in nuclear power plants or deep-sea exploration. The operation of a robotic arm is typically controlled by a computer, which sends commands to the arm's motors to move and rotate the joints. In some cases, the arm may be programmed to perform a specific task autonomously

2.2 OPEN CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using

flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streeview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A fullfeatured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers. OpenCV is cross-platform: developers and users who contribute to the library, provide support, and develop applications using OpenCV. It supports both 2D and 3D vision: OpenCV provides tools and algorithms for both 2D and 3D vision, including depth estimation, stereo vision, and 3D reconstruction. OpenCV has a modular structure: OpenCV is designed with a modular structure, which makes it easy to add and remove modules depending on your needs. It is used in a wide range of applications: OpenCV is used in a wide range of applications, including robotics, augmented reality, medical imaging, surveillance, automotive, and more. OpenCV supports deep learning: OpenCV provides support for It has a large and active community: OpenCv has a large community of popular deep learning frameworks such as TensorFlow, PyTorch, and Caffe, which makes it easy to use deep learning models for tasks such as object detection and image classification. It has a vast collection of tools and functions: OpenCV provides a vast collection of tools and functions for image and video processing, including image filtering, edge

detection, feature detection, object recognition, tracking, and segmentation. OpenCV provides support for multiple cameras: OpenCV provides tools and functions for working with multiple cameras, including camera calibration, stereo vision, and multi-camera tracking.

2.3 Image processing

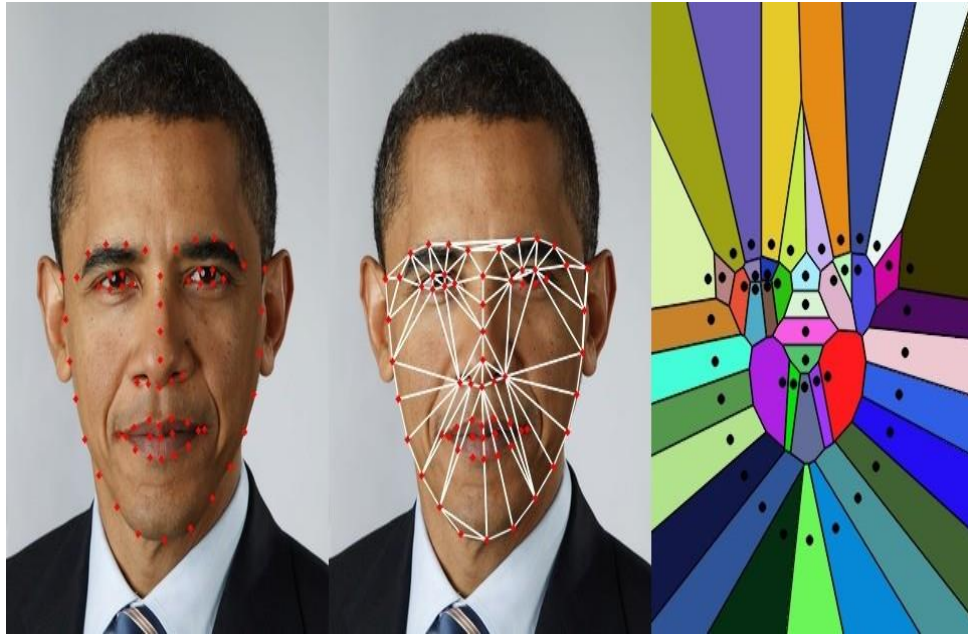


FIG.2.1.IMAGE PROCESSING

FIG.2.1.Says that how image processing works

Types of Image Processing

There are five main types of image processing:

- Visualization - Find objects that are not visible in the image
- Recognition - Distinguish or detect objects in the image
- Sharpening and restoration - Create an enhanced image from the original image
- Pattern recognition - Measure the various patterns around the objects in the image

Components of Image Processing

Computer

A general-purpose computer, which may be anything from a PC to a supercomputer, is used in an image processing system. Sometimes, specifically built computers are utilized in specialized applications to reach a specified degree of performance.

Hardware for Specialized Image Processing

It comprises the digitizer and hardware that can carry out basic operations, including an Arithmetic Logic Unit (ALU), which can carry out simultaneous arithmetic and logical operations on whole pictures.

Massive Storing

In applications involving image processing, the skill is essential. The three main types of digital storage for image processing applications are as follows: Three types of storage exist

- (1) short-term storage,
- (2) online storage for quick recall
- (3) archive storage, which is characterized by rare access.

Camera Sensors

It alludes to perception. The image sensor's primary function is to collect incoming light, transform it into an electrical signal, measure that signal, and then output it to supporting electronics. It consists of a two-dimensional array of light-sensitive components that convert photons into electrons. Images are captured by equipment like digital cameras using image sensors like CCD and CMOS. Two components are often needed on image sensors to collect digital pictures. The first is an actual tool (sensor) that can detect the energy emitted by the object we want to turn into an image. The second is a digitizer, which transforms a physical sensing device's output into digital form.

Software

The image processing software comprises specialized modules that carry out particular functions.

Hardcopy Equipment

Laser printers, film cameras, heat-sensitive equipment, inkjet printers, and digital equipment like optical and CDROM discs are just a few examples of the instruments used to record pictures.

Networking

To send visual data through a networked computer, it is a necessary component. The most important factor in picture transmission is bandwidth since image processing applications require vast amounts of data.

Blind Deconvolution in Image Processing

Blind image deconvolution is the challenge of recovering a clear picture from a blurry and noisy one without knowing precisely how the image was blurred, as obtained by an ideal pinhole camera. The unidentified blurring operation might be brought on by defocus, camera movement, scene motion, or other optical defects. A trade-off between exposure duration and aperture setting is necessary for proper photography exposure. The photographer might utilize a big aperture or a lengthy exposure period when the lighting is bad. The first option produces motion blur when the camera moves concerning objects in the scene while the exposure takes place. When using the second option, things farther from the focal plane become out-of-focus blurred. This may lead to blind deconvolution in image processing.

Before we jump into image processing, we need to first understand what exactly constitutes an image. An image is represented by its dimensions (height and width) based on the number of pixels. For example, if the dimensions of an image are 500 x 400 (width x height), the total number of pixels in the image is 200000.

This pixel is a point on the image that takes on a specific shade, opacity or color. It is usually represented in one of the following:

Grayscale-

A pixel is an integer with a value between 0 to 255 (0 is completely black and 255 is completely white).

RGB-

A pixel is made up of 3 integers between 0 to 255 (the integers represent the intensity of red, green, and blue).

RGBA-It is an extension of RGB with a nadded alpha field, which represents the opacity of the image.

Image processing requires fixed sequences of operations that are performed a

teach pixel of an image. The image process or performs the first sequence of operations on the image, pixel by pixel. Once this is fully done, it will begin to perform the second operation, and so on. The output value of these operations can be computed at any pixel of the image.

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain pre determined signal processing methods.

Components of Image Processing

Computer

A general-purpose computer, which may be any thing from a PC to a super computer, is used in an image processing system. Sometimes, specifically built computers are utilized in specialized applications to reach a specified degree of performance.

Hardware for Specialized Image Processing

It comprises the digitizer and hardware that can carryout basic operations, including an Arithmetic Logic Unit (ALU), which can carry out simultaneous arithmetic and logical operations on whole pictures.

Massive Storing

In applications involving image processing, the skill is essential. The three main types of digital storage for image processing applications are as follows: Three types of storage exist

- (1) short-term storage
- (2) online storage for quick recall
- (3) archive storage, which is characterized by rare access.

Camera Sensors

It alludes to perception. The image sensor's primary function is to collect incoming light, transform it into an electrical signal, measure that signal, and then output it to supporting electronics. It consists of a two-dimensional array of light-sensitive components that convert photons into electrons. Images are captured by equipment like digital cameras using image sensors like CCD and CMOS. Two components are often needed on image sensors to collect digital pictures. The first is an actual tool (sensor) that can detect the energy emitted by the object we want to turn into an image. The second is a digitizer, which transforms a physical sensing device's output into digital form.

Software

The image processing software comprises specialized modules that carry out particular functions.

Hard copy Equipment

Laser printers, film cameras, heat-sensitive equipment, inkjet printers, and digital equipment like optical and CDROM discs are just a few examples of the instruments used to record pictures.

Networking

To send visual data through a networked computer, it is a necessary component. The most important factor in picture transmission is bandwidth since image processing applications require vast amounts of data

Fundamental Image Processing Steps

Image Acquisition

Image acquisition is the first step in image processing. This step is also known as preprocessing in image processing. It involves retrieving the image from a source, usually a hardware-based source.

Image Enhancement

Image enhancement is the process of bringing out and highlighting certain features of interest in an image that has been obscured. This can involve changing the brightness, contrast, etc.

Image Restoration

Image restoration is the process of improving the appearance of an image. However, unlike image enhancement, image restoration is done using certain mathematical or probabilistic models.

Color Image Processing

Color image processing includes a number of color modeling techniques in a digital domain. This step has gained prominence due to the significant use of digital images over the internet.

Wavelets and Multi resolution Processing

Wavelets are used to represent images in various degrees of resolution. The images are subdivided into wavelets or smaller regions for data compression and for pyramidal representation.

Compression

Compression is a process used to reduce the storage required to save an image or the band width required to transmit it. This is done particularly when the image is for use on the Internet.

Morphological Processing

Morphological processing is a set of processing operations form orphing images based on their shapes.

Segmentation

Segmentation is one of the most difficult steps of image processing. It involves partitioning an image into its constituent parts or objects.

Representation and Description

After an image is segmented into regions in the segmentation process, each region is represented and described in a form suitable for further computer processing.

Representation deals with the image's characteristics and regional properties. Description deals with extracting quantitative information that helps differentiate one class of objects from the other.

Recognition

Recognition as signs a label to an object based on its description.

Blind Deconvolution in Image Processing

Blind image deconvolution is the challenge of recovering a clear picture from a blurry and noisy one without knowing precisely how the image was blurred, as obtained by an ideal pinhole camera. The unidentified blurring operation might be brought on by defocus, camera movement, scene motion, or other optical defects. A trade-off between exposure duration and aperture setting is necessary for proper photography exposure. The photographer might utilize a big aperture or a lengthy exposure period when the lighting is bad. The first option produces motion blur when the camera moves concerning objects in the scene while the exposure takes place. When using the second option, things farther from the focal plane become out-of-focus blurred. This may lead to blind deconvolution in image processing.

Importance of Phase in Image Processing

Phase, in a nut shell, contains information about the positions of features. Phase-only and magnitude-only photos cannot be combined to produce the original. To obtain the original, multiply them in the Fourier domain and reverse the transformation. A repeated waveform's phase describes the position or timing of a particular point within a wave cycle. Instead of the actual absolute phases of the signals, the phase difference between waves usually matters.



FIG.2.2.ARDUIUNO

FIG.2.2.Shows that how ardiono looks like.

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contain every thing needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Technical specifications

Microcontroller	Atmega328p
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3VPin	50 mA
Flash Memory	32KB(ATmega328P) Of which 0.5KB used by boot loader
SRAM	2KB(ATmega328P)
EEPROM	1KB(ATmega328P)
Clock Speed	16MHz
LED_BUILTIN	13
Length	68.6mm
Width	53.4mm
Weight	25g

TABLE.2.1TECHNICAL SPECIFICATIONS

The 14 digital input/output pins can be used as input or output pins by using pin Mode(), digital Read() and digital Write() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50K ohms which are disconnected by default.

General pin specifications

- LED: There is a built-in LED driven by digital pin13. When the pin is highvalue, the LED is on, when the pin is low, it is off.
- VIN: The input voltage to the Arduino /Genuino board when

it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack(7-20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- 3V3: A 3.3 volt supply generated by the on-
- Board regulator. Maximum current draw is 50mA.
- GND: Ground pins.
- IOREF: This pin on the Arduino /Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
- Reset: Typically used to add a reset button to shields that block the one on the board.

ARDUINO Hardware part

- Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-A like 2.5 license and are

available on the Arduino website. Layout and production files for some versions of the hardware are also available. Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -uno. Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the on board voltage regulator due to specific form-factor restrictions. Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino UNO is the optiboot boot loader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via

Universal Serial Bus(USB), implemented using USB-to-serial adapter chips such as the FTDI FT232RL. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Board uno, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional

- Micro controller tools, instead of the Arduino IDE, standard AVR in-system Programming (ISP) programming is used. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the under side of the board that can plug into solderless bread boards. Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

ARDUINO Software part

IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program submain() into an executable cyclic executive program.

Hand gestures

Hand gesture recognition provides an intelligent and natural way of human computer interaction (HCI). Its applications range from medical rehabilitation to consumer electronics control (e.g. mobile phone). In order to distinguish hand gestures, various kinds of sensing techniques are utilized to obtain signals for pattern recognition. Acceleration-based and electro myogram-based techniques are two research branches in the field of hand gesture pattern recognition. Acceleration-based (ACC-based) gesture control is usually studied as a supplementary interaction modality. It is well suited to distinguish noticeable, larger scale gestures with different hand trajectories of forearm movements. With ACC-based techniques some subtle finger or hand movement may be ignored where as electro myogram-based (EMG-based) gesture recognition

EDGE DETECTION

Edge Detection is one of the most commonly used operations in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

CANNY

The most significant new dimension to the Canny algorithm is that it tries to assemble the individual edge candidate pixels into contours. These contours are formed by applying a hysteresis threshold to the pixels. This means that there are two thresholds, an upper and a lower. If a pixel has a gradient larger than the upper threshold, then it is accepted as an edge pixel; if a pixel is below the lower threshold, it is rejected. If the pixel's gradient is between the thresholds, then it will be accepted only if it is connected to a pixel that is above the high threshold.



FIG.2.2 HAND GESTURES

FIG.2.2.Says that how hand gestures work.

CHAPTER-3 METHODOLOGY

3.1 Methodology

The first step in developing an Open CV-based robotic arm is to design the mechanical structure of the arm. This involves selecting the type of joints, actuators, and end effectors that will be used, as well as determining the dimensions and range of motion for each joint. The mechanical design should be optimized for the specific application of the robotic arm. Next, Open CV needs to be installed on the computer or embedded system that will be used to control the robotic arm. This can typically be done by following the instructions on the official OpenCV website.

The robotic arm then needs to be connected to the computer or embedded system. This usually involves wiring the motors and sensors to a microcontroller or single-board computer, depending on the specific hardware used.

If vision-based control is desired, the cameras that will be used need to be calibrated. This process involves taking pictures of a calibration pattern from different angles and using OpenCV's calibration functions to compute the intrinsic and extrinsic camera parameters. This calibration is important to ensure accurate vision-based control of the robotic arm.

After the hardware is set up, the software that will control the robotic arm needs to be developed. This software typically involves using OpenCV to detect and track objects, and then using the output of the vision system to generate control signals for the motors. The software needs to be optimized for the specific application of the robotic arm, and may require the use of machine learning algorithms for more advanced applications.

Once the software is developed, the system needs to be tested and refined to ensure that it is working properly. This may involve adjusting the mechanical design or the software algorithms to improve the accuracy and reliability of the system. Additionally, if the robotic arm is part of a larger system, it needs to be integrated with the other

.components using appropriate communication protocols and APIs. Overall, developing an OpenCV-based robotic arm is a complex process that involves a combination of mechanical design, electronics, software development, and testing. The methodology outlined above provides a general framework for developing such a system, but the specific details will depend on the specific application and hardware used.

3.2 Standards

Using MediaPipe and OpenCV to develop a robotic arm controlled by hand gestures is a fascinating project. When it comes to IEEE standards, you'll primarily be concerned with safety, interoperability, and possibly performance benchmarks.

For safety, you might reference standards like IEEE 802.11 for wireless communication protocols, ensuring reliable control signals between the hand gestures and the robotic arm. IEEE 1451 might also be relevant for sensor communication and data interchange standards.

Interoperability standards, such as those outlined in IEEE 11073 for health informatics, could help ensure that your system can communicate effectively with other devices or systems.

For performance benchmarks, you could look at IEEE 754 for floating-point arithmetic, ensuring accurate calculations for the robotic arm's movements based on hand gestures.

Make sure to research and adhere to relevant standards throughout the development process to ensure the quality and reliability of your robotic arm project.

3.3 CONSTRAINTS

Creating a robotic arm controlled by hand gestures using MediaPipe and OpenCV is definitely feasible, but it comes with some constraints to consider:

Accuracy of Hand Gesture Detection: MediaPipe provides hand tracking and gesture recognition capabilities, but the accuracy may vary based on lighting conditions, background clutter, and hand orientation. You'll need to fine-tune your model and possibly implement additional filtering techniques to improve accuracy.

Latency: Real-time control of a robotic arm requires low latency between hand gestures and corresponding actions. Processing the video feed from the camera, detecting gestures, and sending commands to the robotic arm all contribute to latency. Optimizing your code and choosing efficient algorithms can help minimize latency.

Hardware Limitations: The capabilities of your robotic arm hardware may impose constraints on the types of gestures it can interpret and the precision of its movements. Ensure that your chosen hardware can reliably execute the desired gestures without exceeding its mechanical limits.

Safety: Implement safety measures to prevent the robotic arm from causing harm in case of unexpected movements or errors in gesture recognition. This could include software-based limits on movement range, as well as physical barriers or emergency stop buttons.

3.4 Trade off

Complexity: Implementing hand gesture recognition using MediaPipe and OpenCV can be complex, especially if you're not familiar with these libraries. You'll need to understand computer vision principles, image processing, and potentially machine learning algorithms for robust gesture recognition.

Accuracy: The accuracy of hand gesture recognition depends on various factors such as lighting conditions, background noise, and the complexity of gestures. Achieving high accuracy may require extensive training and tuning of the recognition model.

Real-time Performance: Real-time performance is crucial for controlling a robotic arm effectively. Processing hand gestures in real-time using MediaPipe and OpenCV may strain system resources, especially on resource-constrained platforms like embedded systems or mobile devices.

Robustness: The system's robustness is essential to handle different hand sizes, orientations, and gestures reliably. Robustness can be achieved through thorough testing and optimization of the recognition algorithm.

CHAPTER-4

ADVANTAGES AND DISADVANTAGES

Advantages :

- Speedy self-check out/in.
- Improves efficiency.
- Reduces staff daily routine work.
- Theft detection.
- Developing uniform standard.
- Prospect for development.

Disadvantages :

- No global standard.
- High cost.
- Lightning, metal and water interruption.
- Job security.
- Small libraries cannot afford the cost

4.2 Further upgradation

There are several ways to further upgrade an Open CV-based robotic arm to improve its functionality and performance. Here are some ideas:

Add more sensors:

In addition to cameras, other sensors such as force sensors, touch sensors, or proximity sensors can be added to the robotic arm to improve its ability to interact with the environment.

Implement machine learning algorithms:

Machine learning algorithms can be used to train the robotic arm to perform specific tasks more efficiently and accurately. For example, a reinforcement learning algorithm can be used to teach the arm how to pick and place object

CHAPTER -5

RESULT

5.1 Result

An OpenCV-based robotic arm can have several advantages over traditional robotic arms that do not utilize computer vision technology. Here are some potential benefits:

Increased flexibility:

By using computer vision to detect and track objects, the robotic arm can be programmed to pick up a wider range of objects, regardless of their size or shape. This makes the arm more versatile and adaptable to different applications.

Improved precision:

OpenCV provides advanced image processing algorithms that can be used to improve the precision of the robotic arm. This allows the arm to perform tasks that require high accuracy, such as placing objects in precise locations.

Reduced cost:

OpenCV is an open-source library, which means that it is freely available to use and modify. This can help to reduce the cost of developing an OpenCV-based robotic arm, as compared to proprietary vision systems.

Improved safety:

By using computer vision to detect and avoid obstacles, the robotic arm can be programmed to operate more safely around humans and other objects in the environment. Overall, an OpenCV-based robotic arm has the potential to be more versatile, precise, costeffective, and safe than traditional robotic arms that do not utilize computer vision technology.

However, the effectiveness of the robotic arm will depend on the specific hardware and software used, as well as the application requirements.

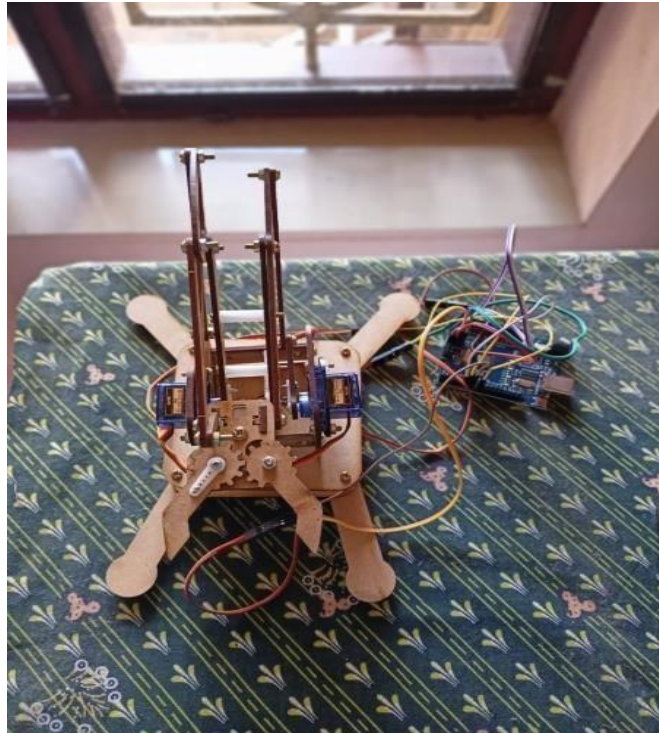


FIG.5.1 CONECTIONS

FIG.5.1.Shows that how connections made



FIG.5.2 ROBOTIC ARM

FIG.5.2. Shows that how robotic arm looks like.

CHAPTER 6

CONCLUSION

In conclusion, an Open CV-based robotic arm has the potential to be a highly effective tool for a wide range of applications, thanks to the versatility, precision, cost-effectiveness, and safety advantages that it offers over traditional robotic arms. By utilizing computer vision technology, the robotic arm can detect and track objects in the environment, allowing it to perform a variety of tasks with increased accuracy and flexibility. However, the success of an Open CV-based robotic arm will depend on the specific application requirements, the hardware and software used, and the quality of the design and implementation. With proper planning, design, and testing, an Open CV-based robotic arm can be a powerful tool for a variety of industries, including manufacturing, healthcare, and logistics.

6.1 FUTURE SCOPE

Creating a robotic arm controlled by hand gestures using MediaPipe and OpenCV has a promising future scope. It could be applied in various fields such as manufacturing, healthcare, and even consumer electronics. Imagine precise control in delicate surgeries or seamless interaction with machines on production lines. As technology advances, the accuracy and versatility of such systems will likely improve, opening up new possibilities for automation and human-machine interaction.

References

- [1] Andre L. C. Barczak, Farhad Dadgostar, "Real-time hand tracking using a set of co-operative classifiers based on Haarlike features", Res. Lett.
- [2] Berry,G.: Small-wall, "A Multimodal Human Computer Intelligent Interaction Test Bed with Applications", Dept. of ECE, University of Illinois at Urbana-Champaign, MS thesis
- [3] B. Stenger, "Template based Hand Pose recognition using multiple cues", In Proc. 7th Asian Conference on Computer Vision : ACCV 2006.
- [4] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, ChuoLing Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. volume abs/1906.08172, 2019.
- [5] C C Wang, K C Wang, "Hand Posture recognition using Adaboost with SIFT for human robot interaction", Springer Berlin, ISSN 0170-8643,
- [6] Elena Sánchez-Nielsen, Luis Antón-Canalís, Mario Hernández-Tejera, "Hand Gesture recognition for Human Machine Interaction", In Proc. 12th International Conference on Computer Graphics, Visualization and Computer Vision : WSCG, 2004
- [7] Eshed Ohn-Bar, Mohan Manubhai Trivedi, "Hand Gesture Recognition In Real-Time For Automotive Interfaces," IEEE Transactions on Intelligent Transportation Systems, VOL. 15, NO. 6, December 2014, pp 2368-2377
- [8] Ju, S., Black, M., Minneman, S., Kimber, D, "Analysis of Gesture and Action in Technical Talks for Video Indexing", IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 97 ,1997

- [9] Lars Bretzner, Ivan Laptev and Tony Lindeberg, "Hand gesture recognition using multiscale colour features, hierarchical models and particle filtering", in Proceedings of Int. Conf. on Automatic face and Gesture recognition, Washington D.C., May 2002
- [10] M. Black, & Jepson, " Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation", In European Conference on Computer Vision, 1996.
- [11] Prajakta Vidhate, Revati Khadse, Saina Rasal, "Virtual paint application by hand gesture recognition system", 2019 International Journal of Technical Research and Applications, 2019, pp. 36-39
- [12] P. Viola and M. Jones, "Robust real-time object detection," Cambridge Res. Lab., Cambridge, MA, pp. 1–24, Tech. Rep. CRL2001/01, 2001.
- [13] Qing Chen , N.D. Georganas, E.M Petriu, "Real-time Vision based Hand Gesture Recognition Using Haar-like features", IEEE Transactions on Instrumentation and Measurement -2007
- [14] Q. De Smedt, H. Wannous and J. Vandeborre, "Skeleton Based Dynamic Hand Gesture Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 1206-1214, doi: 10.1109/CVPRW.2016.153.
- [15] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in Proc. IEEE Int. Conf. Image Process., 2002, vol. 1, pp. 900–903.
- [16] Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010
- [17] Zhag F, Bazarevsky, Vakunov A et.al, MediaPipe Hands: On – Device Real Time Hand Tracking, Google Research. USA. 2020.
<https://arxiv.org/pdf/2006.10214.pdf>.