OXFORD COLLEGE,HUBLI

 Bachelor Of Computer Application

Major Project Report

“HAND GESTURE RECOGNITION USING MACHINE“

By

VAISHNAVI SINNUR(20U10442)

Under the supervision of

Prof. Manjunath G.M

ASSOCIATE PROFESSOR, DEPARTMENT OF ‘BCA’

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION(BCA)

OXFORD COLLEGE,HUBLI

**OXFORD COLLEGE HUBLI**

****



DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION(BCA)

Keshwapur, Hubli - 580023

Karnataka, India

**CERTIFICATE**

This is to Certify That the project Report Entitled

“HAND GESTURE RECOGNITION USING MACHINE“

SUBMITTED TO THE PARTIAL FULFILLMENT FOR THE

AWARD OF THE DEGREE

BACHELOR OF COMPUTER APPLICATION

Is A Result of the Bonafide Work Carried by

MS. VAISHNAVI SINNUR

(20U10442)

**During Academic Year 2022-2023**

UNDER THE GUIDANCE OF

Prof. Manjunath G.M

***ACKNOWLEDGEMENT***

***“ Gratitude makes sense of our past, brings peace for today and creates a vision for tomorrow”.***

***So, we expressed our gratitude to all those people without whose support, encouragement, guidance and co-operation this project would not have been completed***

***We would like to take privilege to thank all those who have guided and inspired us in completion of the project.***

***Firstly, we would like to express my heartfelt thanks to our respected Principal prof. Manjunath Muttalgerisir for his keen interest and valuable inputs throughout the course of this work.***

***Secondly, we are thankful to our respected Teacher Prof. Manjunath Muttalgeri for giving us opportunity to carry out the project.***

***Last but not least I sincerely thank my beloved parents and family who were very cooperative and supported us morally during the course***

**ABSTRACT**

Machine learning is the branch of AI which is focused on building applications

which can learnfrom user input and increase their accuracy over a period of time without being programmed.In this project we train a machine learning model such that, It allows us to capture the gesturesmade by the human or users and provide the desired output.

Hand gesture recognition for human-computer interaction is an area of active research

In computer vision and machine learning. The primary goal of gesture recognition research is to create a system, which can identify specific human gestures and use them to conveyinformation or for device control.

Hand Gesture recognition, although has been exploring for many years, is still a challenging problem. Complex background, camera angles and illumination conditions make the problem more difficult. Thus, this paper presents a fast and robust method for hand gesture recognition based on RGB video. First we detect the skin based on their color. Then we extract the contour and segment the hand region. Finally we recognize the gesture. The results of experiment demonstrate that the proposed method are efficient to recognize gesture with a higher accuracy than the state of the art.

CONTENTS

* Abstract………………………………………………………………..v
* List of Abbreviation…………………………………………………...vii
* List of Figures………………………………………………………….vii

1) Chapter 1: Introduction

1.1 Purpose

1.2 Scope

1.3 Problem statement

1.4 Controls and respective gestures

1.6 Media player

2) Chapter 2: Literature Review ……………………………………………………………......... 7

3) Chapter 3: Requirements………………………………………………………………………… 9

3.1) Functional requirements

3.2) Non-Functional requirements

3.3) Software and Hardware requirements

4) Chapter 4:Methodology……...……………………………………………………………......... 11

4.1) Design

4.2) System data flow diagram

4.3) Proposed scheme for gesture spotting

5) Chapter 5: Implementation……………………………………………………………………… 15 6) Chapter 6: Testing and results…………………………………………………………………… 19 7) Chapter 7: Conclusion and Future work………………………………………………………… 24 8) Reference……………………………………………………………………………………….. 26

LIST OF ABBREVIATIONS

These are the some abbreviations we used in our project.

| ML | Machine Learning |
| --- | --- |
| AI | Artificial Intelligence |
| LSTM | Long Short Term Memory |
| DL | Deep Learning |
| API | Application Programming Interface |
| Open CV | Open Source Computer Vision |
| ANN | Artificial Neutral Network |

**LIST OF FIGURES**

|  | 5 Finger (FORWARD) |  |
| --- | --- | --- |
| 4.1 | Design | 12 |
| 4.2 | System Data Flow Diagram | 13 |
| 4.3 | Proposed Scheme for Gesture Spotting | 14 |
| 6.1 | 1 Finger(MUTE) | 21 |
| 6.2 | 2 Finger(PLAY / PAUSE) | 21 |
| 6.3 | 3 Finger(VOLUME UP) | 22 |
| 6.4 | 4 Finger(VOLUME DOWN) | 22 |
| 6.5 | 5 Finger(FORWARD) | 23 |

**CHAPTER 1**

**INTRODUCTION**

Chapter 1

Introduction

Hand gestures are an aspect of body language that can be conveyed through the center of the palm, the finger position and the shape constructed by the hand. Hand gestures can be classified into static and dynamic. As its name implies, the static gesture refers to the stable shape of the hand, whereas the dynamic gesture comprises a series of hand movements such as waving. There are a variety of hand movements within agesture for example, a handshake varies from one person to another and changes according to time and place. The main difference between posture and gesture is that posture focuses more on the shape of the hand whereas gesture focuses on the hand movement. The main approaches to hand gesture research can be classified into wearable glove-based sensor approach and the camera vision-based sensor approach.

Gesture recognition is a technique which is used to understand and analyze the human bod

Language and interact with the user accordingly. This in turn helps in building abridge between the machine and the user to communicate with each other. Gesture recognition is useful in processing the information which cannot be

conveyed through speech or text. Gestures are the simplest means of communicatin something that is meaningful. This paper involves implementation of the system that aims to design a vision-based hand gesture recognition system with a high correct detection rate along

with a high-performance criterion, which can work in a real time Human ComputerInteraction system

without having any of the limitations (gloves, uniform background etc.) on the user environment.

There has been great emphasis on Human-Computer-Interaction research to create

easy-to-use interfaces by directly employing natural communication andmanipulation skills of humans. As an important part of the body, recognizing handgesture is very important for Human- Computer-Interaction. In recent years, there has been a tremendous amount of research

been a tremendous amount of research on hand gesture recognition.

While there are numerous researches focused on this topic, there are still several problems to

and fast method is needed to improve user experiences. A contour based method for hand gesture using depth image data is shown in [3]. Using depthdata, thesemethod can distinguish the hand from background easily without getting confused bybackground color. However, depth data are not common and easily available while RGB data solves the problem. In a hierarchical method of static hand gesturerecognition that combines finger detection and histogram of oriented gradient (HOG)feature is proposed. An algorithm applied for locating fingertips in hand regionextracted by Bayesian rule based skin color segmentation is proposed. In the continuous gesture recognition problem is tackled with a tow streams management. We have assumed that the trained model should have greater accuracy to avoid misclassification. The model selection is totally dependingupon model performance and prediction score. Purpose is to control mediaplayer using Hand Gestures.

learning.

* 1. Purpose:

The gestures are chosen because they are commonly used to communicate andcan thus

be used in various applications such as, media player, stockmanagement. We have assumed that the

trained model should have greateraccuracy to avoid misclassification. The model selection is totally

dependingupon model performance and prediction score. Purpose is to control mediaplayer using

Hand Gestures.

1.2) Scope: The scope of this project is to create a method to recognize handgestures based on a machine learning technique, employing histograms of local orientation, the orientation histogram will be used as a feature vector for gestureclassification. And interpolation. High priority for the system is to be simplewithout making use of any special hardware. All the competition should occur ona workstation or PC. Special hardware would be used only to digitize the image(Scanner or digital camera).

1.3) Problem Statement: “Controlling media player using hand gestures, withthe help of Machine learning and open cv”

1.4) CONTROLS AND RESPECTIVE GESTURES:

Medial player controls are Play, Pause, Volume up, Volume down Forward, backward. Next, previous and mute. In this project we are using certain hand gestures to control the media player.Like, 2 finger gesture Represents play or pause functions three finger hand gesture represents volume up control, whereas four finger and gesture represents volume down in control,five finger represents forward fist represents the backward thumbs up will be the next video in the list and thumbs down will be previous video in the list, andone finger represents the mute.

| CONTROLS | GESTURES |
| --- | --- |
| Play / pause | 2 finger |
| Volume up | 3 fingers |
| Volume down | 4 fingers |
| Forward | 5 fingers |
| Backward . | These above Fist |
| Next | Thumbs up |
| Previous | Thumbs down |
| Mute | 1 finger |

These above gestures have implemented,by showing these gestures to camera while the model is running we can handle the media player.

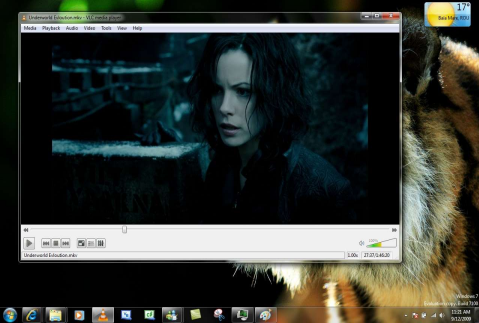
1.5) Open CV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision andmachine learning software library. OpenCV was built to provide a common infrastructure forcomputer vision applications and to accelerate the use of machine perception in thecommercial products. Being a BSD-licensed product, OpenCV makes it easy for businessesto utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set ofboth classic and state-of-the-art computer vision and machine learning algorithms. Thesealgorithms can be used to detect and recognize faces, identify objects, classify human actionsin 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 highresolution image of an entire scene, find similar images from an image database, remove redeyes from images taken using flash, follow eye movements, recognize scenery and establishmarkers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand peopleof user community and estimated number of downloads exceeding 18 million. The library isused 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 spanthe range from stitching streetview images together, detecting intrusions in surveillance videoin Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, runninginteractive art in Spain and New York, checking runways for debris in Turkey, inspectinglabels 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 andMac OS. OpenCV leans mostly towards real-time vision applications and takes advantage ofMMX and SSE instructions when available. A full - featured and interfaces are being activelyC

developed right now. There are over 500 algorithms and about 10 times as many functions thatcompose or support those algorithms. OpenCV is written natively in C++ and has a templatedinterface that works seamlessly with STL containers. 5

1.6) Media player:



In this project we are using VLC media player which is very much feasible to control with the

keyboard. While running this model we recommend to use this media player or any other

media player which have same specifications. VLC is a free and open source

cross-platform multimedia player and framework that plays most multimedia files, and

various streaming purpose .

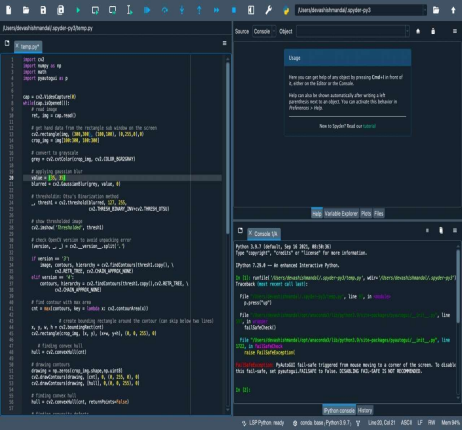
VLC media player (previously the VideoLAN Client and commonly known as simply VLC) is a free and open-source, portable, cross-platform media player software and

streaming media server developed by the VideoLAN project. VLC is available for

In this project we are using VLC media player which is very much feasible to control

desktop operating systems and mobile platforms, such as Android, iOS and iPadOS. 6

1.7) Interpreter



In this project we are using SPYDER interpreter, Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

7

**CHAPTER 2**

**LITERATURE REVIEW**

Chapter 2

Literature Review

To understand the state of the art work in the area of hand gesture recognition using

machine learninig,we conducted a literature survey.The summary has been given in

the table

| Author’s Name/ Paper Title | Year | Technology /Design | Result shared by author | What you infer? |
| --- | --- | --- | --- | --- |
| 1.Suharjito,H.  Gunawan,N.  Thiracitta and A.  Nugroho. | ‘2018’ | Video-based sign  Language  Recognition | Recognition of a sign not only by the shape but also by the action the signer does | Model that  Classifies video  clips based on  sequence of  frames. |
| 2.Vijay Shinde, TusharBacchav, Jitendra Pawar  and Mangesh  Sanap. | ‘2018’ | Human Computer  Interaction (HCI) | Using library ‘pyautogui’ which works as an API,  between ML model and the pc keyboard. | API that connects the keyboard and  ML model. |
| 3.”Machine  Learning  Techniques for  Indian Sign  Language  Recognition," | ‘2017’ | MATLAB | Classification of single and double handed Indian sign language recognition alogorithms with the help of MATLAB with 92-100% of accuracy. | Recognizing  different sign  language using  machine learning. |
| Rohini M,  Abhishek Leo  Kingston. j,  Shriram G S,  Siva Sankaran & Vasuki G, | ‘2018’ | Open CV | Open CV is the place where one can get pre  processed data, Training machine learning model is not necessary. | Using Open-cv makes us to  prepare machine  learning model  easilysince,we  get pre processed data. |

place With the help of literature survey, we came across certain libraries like PyAutoGUI which actsas an API that controls the keyboard and the mouse. Along with that we came to the conclusion that desigining machine learning midel using Open CV, training the model is not necessary.

CHAPTER

REQUIREMENTS

Chapter 3

Requirements

**3.1) Functional Requirements:**

Gesture analysis using machine learning has various applications, this project deals with the • The camera used will be able to capture user images from the real time video sequences. • The media player used will be able to handle it’s controls with the keyboard (eg: VLCMedia player). • The interpreter used will be able to have access for controlling camera and keyboard (eg:

3.2) Non Functional Requirements:

• The distance with which we show the hand gestures should be less than 2 meters.

• Background should be plain whenever hand gesture is shown.

3.3) Software and Hardware requirements:

**Hardware Requirements**:

Processor : Any Processor above 500 MHz

controlling of media player using hand gestures.

Ram : 4GB Hard Disk : 512GB

Input Device : Standard Keyboard AND Mouse.

Output Device : High display monitor.

**Software Requirements:**

Microsoft Windows XP or later / Ubuntu 12.0 LTS or later /MAC OS 10.1 or later. Python Interpreter (3.6).Python -IDLE(Python 3.4 64 bit) or any python softwareTensorFlow framework, Keras API , AND Open CV.

CHAPTER 4

METHODOLOGY

Chapter 4

Methodology

Whenever we provide hand gesture to the camera while model is running, Itcaptures the image and convert it to the grey scale image for the imagepreprocessing. Wherein image pre processing feature extraction will happen. Thatextracted data is used to recognize the gesture given by the user respective taskwill be performed.

Recognized hand

gesture

Gesture

Performing

Extracted data

recognition

specific task

system

Fig~~ure 4.1 - Design~~

Feature

extraction

Pre-processed

image

Image

Hand movement Capture video

processing

image Digital camera

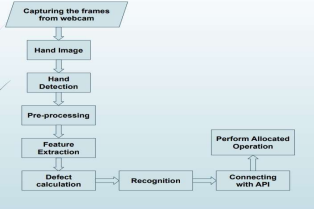
12

Figure 4.2 - System Data Flow Diagram

Once the hand is subjected to the camera, then it captures the frames of real timeimages. Then it converts to grayscale image. After the detection phase pre-processinghappens, the grayscale image is subjected into thresholding, wherein only thespecified image pixels from the range 127 - 255 remain. And then the other pixelsare converted to black. From The convexity hull we would be trying to find out contours that is exactly thecontours formed between finger angles. If the angle between the fingers is <= 90 degrees we will be incrementing the value ofcontours in the count variable, according to the number of count value captured wewould be bringing out actions onto the media player. Action to the media player from gestures is done by the python library PyAutoGUI,

where it acts as an API between the model and the keyboard.

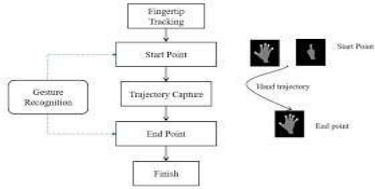


Figure 4.3 - Proposed scheme for gesture spotting.

A hand gesture recognition system was developed to capture the hand gestures beingperformed by the user and to control a computer system based on the incominginformation. Many of the existing systems in literature have implemented gesturerecognition using only spatial modelling, i.e. recognition of a single gesture and not temporal modelling i.e. recognition of motion of gestures. Also, the existing systemshave not been implemented in real time, they use a pre captured image as an input forgesture recognition.

CHAPTER 5

IMPLEMENTATION

Chapter 5

Implementation

**Packages imported**

import numpy as np

import pyautogui as p

**Converting BGR to Gray scale images**

**> Reads image**

ret, img = cap.read()

**> Get hand data from the rectangle sub window on the screen**

Cv2.rectangle(img,(300,300),(100,100),(0,255,0),0)

crop\_img = img[100:300, 100:300]

**> Convert to grayscale**

grey = cv2.cvtColor(crop\_img, cv2.COLOR\_BGR2GRAY)

`

**> Applying gaussian blur**

value = (35, 35)

blurred = cv2.GaussianBlur(grey, value, 0)

**> Thresholding: Otsu's Binarization method \_,**

thresh1 = cv2.threshold(blurred, 127, 255,

cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU)

**> Show thresholded image**

cv2.imshow('Thresholded', thresh1)

**>Check OpenCV version to avoid unpacking error**

(version, \_, \_) = cv2.\_\_version\_\_.split('.')

16

import cv2

import math a

>

**> Find contour with max area**

cnt = max(contours, key = lambda x: cv2.contourArea(x))

**> Create bounding rectangle around the contour (can skip below two lines)**

x, y, w, h = cv2.boundingRect(cnt)

cv2.rectangle(crop\_img, (x, y), (x+w, y+h), (0, 0, 255), 0)

**> Finding convex hull**

hull = cv2.convexHull(cnt)

**> Drawing contours**

drawing = np.zeros(crop\_img.shape,np.uint8)

cv2.drawContours(drawing, [cnt], 0, (0, 255, 0), 0)

cv2.drawContours(drawing, [hull], 0,(0, 0, 255), 0)

**> Finding convexity defects**

defects = cv2.convexityDefects(cnt, hull)

count\_defects = 0

cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3)

**> Applying Cosine Rule to find angle for all defects (between fingers) with angle > 90 degrees and ignore defects**

for i in range(defects.shape[0]):

s,e,f,d = defects[i,0]

start = tuple(cnt[s][0])

end = tuple(cnt[e][0])

far = tuple(cnt[f][0])

**> Find length of all sides of triangle**

a = math.sqrt((end[0] - start[0])\*\*2 + (end[1] - start[1])\*\*2)

b = math.sqrt((far[0] - start[0])\*\*2 + (far[1] - start[1])\*\*2)

c = math.sqrt((end[0] - far[0])\*\*2 + (end[1] - far[1])\*\*2)

**> Apply cosine rule here**

angle = math.acos((b\*\*2 + c\*\*2 - a\*\*2)/(2\*b\*c)) \* 57

**> Ignore angles > 90 and highlight rest with red dots**

if angle <= 90:

count\_defects += 1

cv2.circle(crop\_img, far, 1, [0,0,255], -1)

#dist = cv2.pointPolygonTest(cnt,far,True)

**> Define actions required**

if count\_defects == 1:

p.press("space")

cv2.putText(img,"2 finger,Space", (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 2, 2)

elif count\_defects == 2:

str = "3 fingers,VOLUP"

p.press("up")

cv2.putText(img, str, (5, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 1, 2)

elif count\_defects == 3:

cv2.putText(img,"4 fingers,VOLDWN", (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX,2,2

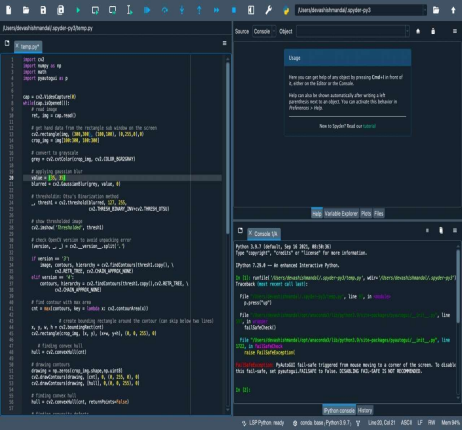
elif count\_defects == 4:

cv2.putText(img,"5 fingers,FWD", (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 2, 2)

else:

cv2.putText(img,"entire hand", (50, 50),\

cv2.FONT\_HERSHEY\_SIMPLEX, 2, 2).



CHAPTER 6

TESTING AND RESULTS

Chapter 6

Testing and Results table

Testing and Results

Pass Pass The following diagram represents the testing and result cases, We used certain controls like play, pause, volume up, volume down, forward, backward, and mute. For these controls we have used hand gestures like 2 fingers, 3 fingers, 4 fingers, 5 fingers, fist, 1 finger respectively.

| Controls | Gestures | Keyboard key | Expected output | Output we  got | Pass/fail |
| --- | --- | --- | --- | --- | --- |
| Play / pause | 2 Fingers | Space | Pause / Play | Pause / play | Pass |
| Volume up | 3 fingers | Up arrow | Vol-up | Vol-up | Pass |
| Volume  Down | 4 Fingers | Down  arrow | Vol-down | Vol-down | Pass |
| Forward | 5 Fingers | Right arrow | Fwd | Fwd | Pass |
| Backward | Fist | Left arrow | Bwd | Bwd | Pass |
| Mute | 1 Finger | M Key | Mute | Mute | Pass |

These are the some screenshots taken while testing and running our project in SPYDER

Interpreter.

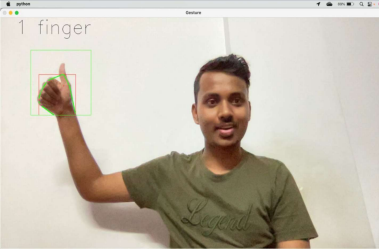


Figure 6.1 – 1 finger (MUTE)

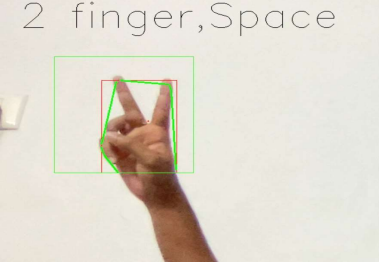


Figure 6.2 - 2 finger (PLAY / PAUSE) 21

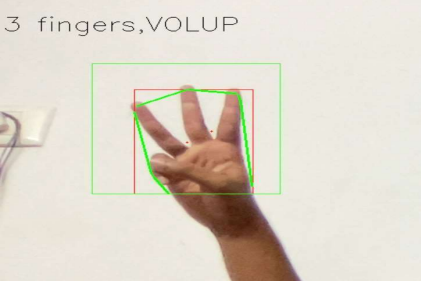


Figure 6.3 – 3 finger (VOLUME UP) 0



Figure 6.4 – 4 finger (VOLUME DOWN) 22



Figure 6.5 – 5 finger (FORWARD) 23

CHAPTER 7

CONCLUSION

Chapter 7

Conclusion and

Future work

This project is designed to analyze the hand gestures made by the user and produce theoutput and here we are using different gestures in order to control the media player. Herethe machine learning model will be analyzing the gesture made by the user and toproduce the required output, as gesture analysis and machine learning has lot moreapplications and future scopes which can be applied in wide range of applications.

* The same concept can also be used for stock management
* It can also be used to convert voice messages to texts
* It can also be used to to convey the useful informations to the robots.

References:

1. Suharjito, H. Gunawan, N. Thiracitta and A. Nugroho, "Sign Language Recognition UsingModified Convolutional Neural Network Model," 2018 Indonesian Association for PatternRecognition International Conference (INAPR), Jakarta, Indonesia, 2018, pp. 1-5, doi:

10.109/INAPR.20188627014.

2. G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," 2018 Conference on Signal Processing AndCommunication Engineering Systems (SPACES), Vijayawada, 2018, pp. 194-197, doi:

10.1109/SPACES.2018.8316344

3. Sign Language Translator Application Using OpenCV Published under licence by IOPPublishing Ltd IOP Conference Series: Materials Science and Engineering, Volume 333, International Conference on Advanced Materials for Better Future 2017 4–5 September 2017, Surakarta, Indonesia

4. K. K. Dutta and S. A. S. Bellary, "Machine Learning Techniques for Indian Sign LanguageRecognition," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, 2017, pp. 333-336, doi:

10.1109/CTCEEC.2017.8454988

.

5. Vijay Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap. 2014. International Journal of Engineering Research & Technology (IJERT). ISSN NO: 2278-0181 Vol. 3 Issue 1,

January-2014

6. Rohini M, Abhishek Leo Kingston. j, Shriram G S, Siva Sankaran & Vasuki G, “HandGesture Recognition Using OpenCV”, International Journal of Scientific Research in ScienceTechnology (IJSRST). ISSN NO:2395-602X, DOI:10.32628, March-April 2021.

.

7. Mr. Deepak K. Ray, Mayank Soni –Hand Gesture Recognition using Python / International Journal on Computer Science and Engineering (IJCSE) . Assistant Professor: MCA Dept , ICEM, Parandwadi Pune, India.

.

APPENDIX A

OPEN CV

OpenCV is the huge open-source library for computer vision. By using it, one canprocess images and videos to identify objects, faces, or even the handwriting of ahuman. When integrated with various libraries, such as NumPy, python is capable ofprocessing the OpenCV array structure for analysis. To Identify image pattern and itsvarious features we use vector space and perform mathematical operations on these features.

APPENDIX B

PyAutoGUI

PyAutoGUI is a Python module which can automate your GUI and programmatically controlyour keyboard and mouse. This article illustrates the GUI functions to create display boxes.PyAutoGUI has ability to control Mouse and Keyboard Automation. PyAutoGUI does not comewith python, so go to command prompt and type the following: pip3 install PyAutoGUI.