**SONOPANT DANDEKAR SHIKSHAN MANDALI’S**



A

Project Report On

**VIRTUAL DRAG & DROP SYSTEM USING OPENCV & PYTHON**

SUBMITTED BY

**MR. KARTIK POOJARI**

**66033**

UNDER THE GUIDANCE

**Prof. TEJAL PATIL**

THIRD YEAR BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

SEMESTER-V

**MUMBAI UNIVERSITY**

**2021-2022**

**SONOPANT DANDEKAR SHIKSHAN MANDALI’S**



**SONOPANT DANDEKAR ARTS, V.S. APTE COMMERCE AND M.H.**

**MEHTA SCIENCE COLLEGE,**

**PALGHAR-401404.**

**Department of Computer Science.**

**CERTIFICATE**

This is to certify that **KARTIK POOJARI** is a student studying in TY.BSC.CS SEM V. She has completed project work entitled **VIRTUAL DRAG AND DROP SYSTEM USING OPENCV & PYTHON** under the guidance of Faculty Member **Asst** **Prof. TEJAL PATIL** satisfactorily and has submitted to the University of Mumbai in partial fulfilment of the requirement during the academic year 2021-2022. The matter presented in the project report has not been submitted earlier.

**College Stamp**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**H.O.D Project Guide External**



PLAGIARISM SCAN REPORT

Words 1973 Date October 22, 2021

Characters 15155 Excluded URL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0%  Plagiarism | 100%  Unique | 0  Plagiarized Sentences |  | 7  Unique  Sentences |

Content Checked For Plagiarism

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **CONTENT** | **PAGE NO.** |
| 1. | Acknowledgment | 5 |
| 2. | Requirement | 6 |
| 3. | Abstract | 7 |
| 4. | Introduction | 8 |
| 5. | Related Works | 9 |
| 6. | Objectives and Features | 10 |
| 7. | System  Architecture and Working | 11 |
| 9. | Coding | 12-13 |
| 10. | Result | 14 |
| 12. | Conclusion | 15 |
| 13. | Future Scope | 16 |
| 14. | Challenges | 17 |
| 15. | Reference | 18 |
|  |  |  |

**ACKNOWLEDGEMENT**

The gratification and joy that accompanies the successful completion of any task would be incomplete without the humble and deep-felt expression of gratitude to everyone who has made it possible.

I would like to extend my heartfelt gratitude to **Asst Prof.**  **RASHMI VARADE** who gave me an opportunity to work under her guidance in this department.

Further would like to place on record my grateful thanks to friends and family who spent their valuable time in guiding me in this project, whose cooperation and encouragement, has immensely helped in the completion of the project.

I also would like to convey my sincere thanks to **Asst Prof. TEJAL PATIL** my internal project guide, who has been a constant source of motivation and inspiration and has helped in bring out the best in me. I am also thankful to all staff members of computer science department who as a team have contributed in the successful completion of this project.

Place: Palghar Date:

**Kartik Poojari**

**REQUIREMENT**

1. Hardware requirement:
   * Laptop(4GB RAM)
   * i3/i5 Processor
   * 256GB/1TB hard disk
   * Camera/Webcam(4-8 megapixel)

1. Software requirement:
   * Python IDLE 3.9(64 bits)
   * Pycharm Professional
   * Cvzone package(ver. 1.4.1)
   * Numpy (Library For Array Processing Package)
   * Mediapipe package (ver. 0.8.7)
   * HandDetector (Library For Hand Detection)
   * Open CV (Machine Learning Software Library)  Tenser Flow

**ABSTRACT**

In modern technologies video tracking and processing the feed has been very essential. This processed data can be used for many research purposes or to express a particular output on a particular system. There are various methods for processing and manipulation of data to get the required output. This VirtualDragDrop application is created using OpenCV module and python programming language which is an apex machine learning tool to create an application like this. Given the real time webcam data, this VirtualDragDrop python application uses OpenCV library to track an object-of-interest (a bottle cap in this case) and allows the user to Drop by moving the object, which makes it both awesome and challenging to Drag simple things.

Keywords: Machine learning, OpenCV, Morphing Techniques, Human-Computer Interactions, Air Writing.

**I. INTRODUCTION**

OpenCV was launched in August 1999 at the Computer Vision and Pattern Recognition conference (and so turns 17 years old at the publication of this book). Gary Bradski founded OpenCV at Intel with the intention to accelerate both the research and use of real applications of computer vision in society. OpenCV has nearly 3,000 functions, has had 14 million downloads, is trending well above 200,000 downloads per month, and is used daily in millions of cell phones, recognizing bar codes, stitching panoramas together, and improving images through computational photography.

OpenCV is at work in robotics systems—picking lettuce, recognizing items on conveyor belts, helping self-driving cars see, flying quad-rotors, doing tracking and mapping in virtual and augmented reality systems, helping unload trucks and pallets in distribution centres, and more—and is built into the Robotics Operating System (ROS).It is used in applications that promote mine safety, prevent swimming pool drownings, process Google Maps and street view imagery, and implement Google X robotics, to name a few examples.

OpenCV has been re-architected from C to modern, modular C++ compatible with STL and Boost. The library has been brought up to modern software development standards with distributed development on Git. Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. OpenCV is a computer vision and machine learning software library that includes many common image analysis algorithms that will help us build custom, intelligent computer vision applications. In this application frequent image feed results in video tracking of our particular object of interest. Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses humancomputer interaction, security and augmented reality, traffic control, medical imaging, and video editing.

**II.**  RELATED WORK

Machine Learning applications are becoming an integral part of our life and for the surrounding society. Real life applications in the field of machine learning can improve the efficiency and consistency in our day to day work, by using computer vision and its associating algorithms and its subsequent modules or libraries can help us develop this type of real life applications.

OpenCV library is an open source library for building a machine learning architecture that can track and process real life data and manipulate it for a required set of expected output.

Related research on various papers and its summaries are as follows:

Title of the Paper is Hand Gesture Recognition its authors are Rafiqul Zaman Khan, Noor Adnan Ibraheem and the related paper’s essential aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a robot or conveying meaningful information. How to form the resulted hand gestures to be understood and well interpreted by the computer considered as the problem of gesture interaction Human computer interaction (HCI) also named Man-Machine Interaction. Title of the Paper is Gesture Controlled Computer

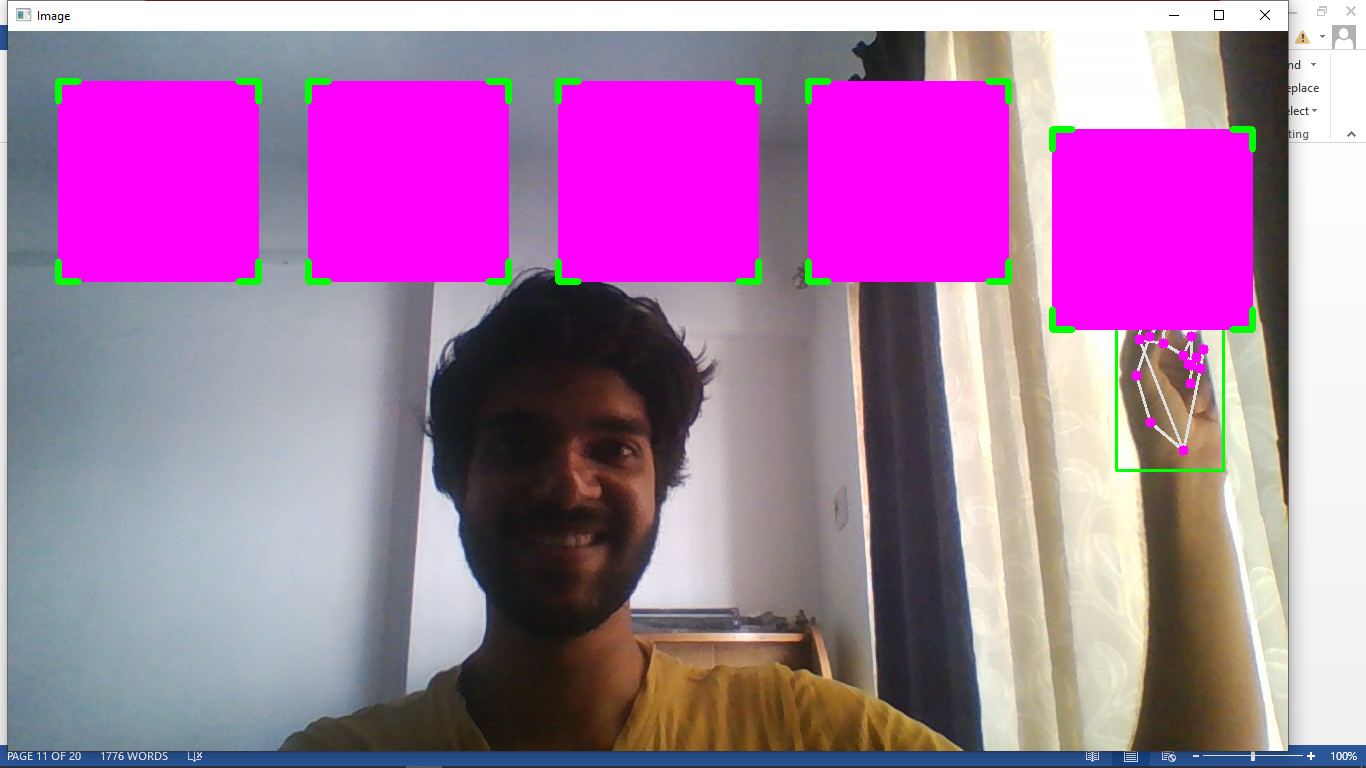
its authors are S. T. Gandhe, Nikita A. Pawar, Mayuri S. Hingmire, Kalpesh P. Mahajan, Devshri V. Patil. This paper describes the method for humans to interact with digital world and use the computer with just our hand movements. The paper is based on image processing. The camera detects gestures and converts those gestures into equivalent digital algorithms as programmed. This paper deals with the controlling all operations of mouse such as right click, left click and movement of cursor over the desktop, drag and drop, snapshot, Air writing and painting through hand gestures.

**OBJECTIVES AND FEATURES**

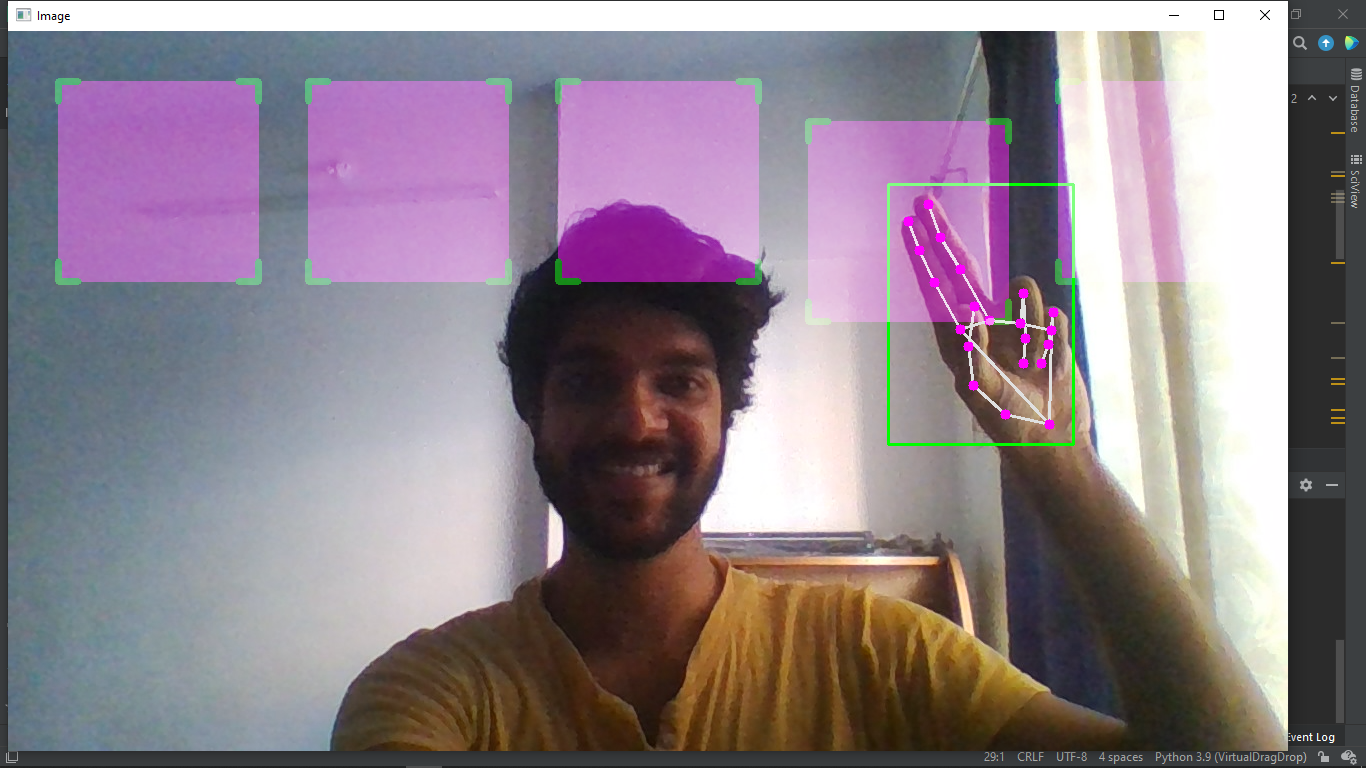
* Move the Hand to the object.
* Press, and hold down, the hand detection module or other pointing device, to "grab" the object.
* "Drag" the object to the desired location by moving the handgesture to this one.
* "Drop" the object by releasing the two-finger of hands.

**SYSTEM ARCHITECTURE AND WORKING**

Coding Part 1:- Draw solid, In which fingers didn’t show when you drag the object to present yourself.



Coding Part 2:- Draw Transparency, In which fingers are visible when you drag the object to present yourself.



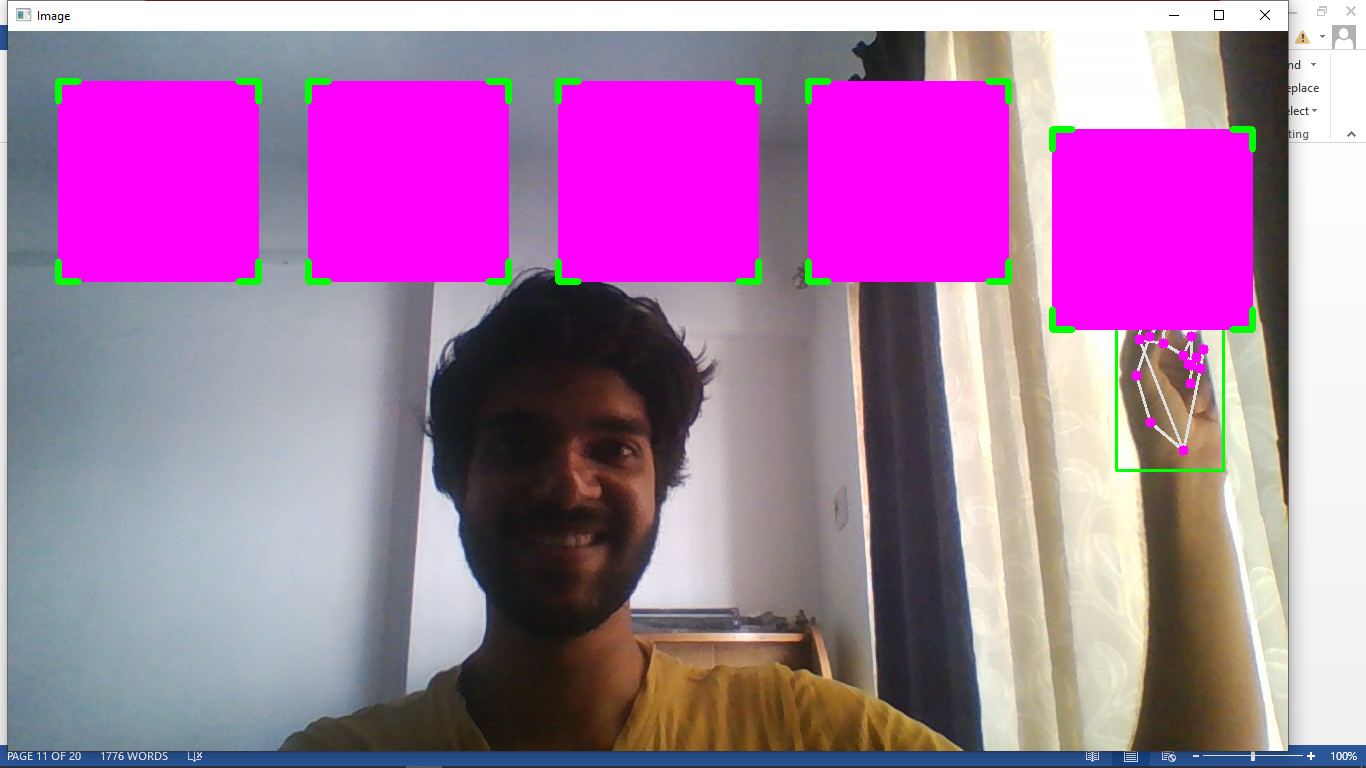
**CODING**

import cv2  
from cvzone.HandTrackingModule import HandDetector  
import cvzone  
import numpy as np  
  
cap = cv2.VideoCapture(0)  
cap.set(3, 1280)  
cap.set(4, 720)  
detector = HandDetector(detectionCon=0.8)  
colorR = (255, 0, 255)  
  
cx, cy, w, h = 100, 100, 200, 200  
  
  
class DragRect():  
 def \_\_init\_\_(self, posCenter, size=[200, 200]):  
 self.posCenter = posCenter  
 self.size = size  
  
 def update(self, cursor):  
 cx, cy = self.posCenter  
 w, h = self.size  
  
 # If the index finger tip is in the rectangle region  
 if cx - w // 2 < cursor[0] < cx + w // 2 and \  
 cy - h // 2 < cursor[1] < cy + h // 2:  
 self.posCenter = cursor  
  
  
rectList = []  
for x in range(5):  
 rectList.append(DragRect([x \* 250 + 150, 150]))  
  
while True:  
 success, img = cap.read()  
 img = cv2.flip(img, 1)  
 img = detector.findHands(img)  
 lmList, \_ = detector.findPosition(img)  
  
 if lmList:  
  
 l, \_, \_ = detector.findDistance(8, 12, img, draw=False)  
 print(l)  
 if l < 30:  
 cursor = lmList[8] # index finger tip landmark  
 # call the update here  
 for rect in rectList:  
 rect.update(cursor)  
  
 ## Draw solid  
 for rect in rectList:  
 cx, cy = rect.posCenter  
 w , h = rect.size  
 cv2.rectangle(img, (cx - w // 2, cy - h // 2),  
 (cx + w // 2, cy + h // 2), colorR, cv2.FILLED)  
 cvzone.cornerRect(img, (cx - w // 2, cy - h // 2, w, h), 20, rt=0)

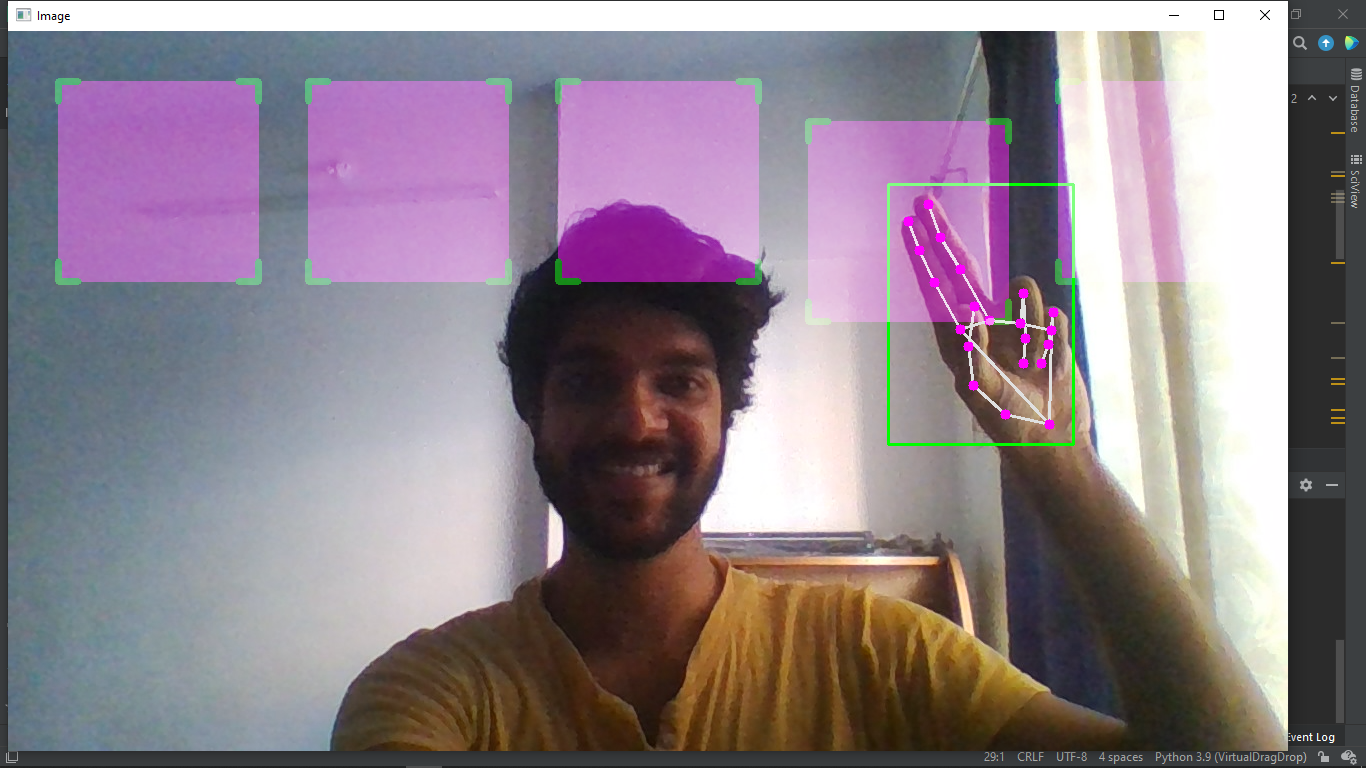
## Draw Transperency  
 # imgNew = np.zeros\_like(img, np.uint8)  
 # for rect in rectList:  
 # cx, cy = rect.posCenter  
 # w, h = rect.size  
 # cv2.rectangle(imgNew, (cx - w // 2, cy - h // 2),  
 # (cx + w // 2, cy + h // 2), colorR, cv2.FILLED)  
 # cvzone.cornerRect(imgNew, (cx - w // 2, cy - h // 2, w, h), 20, rt=0)  
  
 out = img.copy()  
 alpha = 0.5  
 mask = img.astype(bool)  
 out[mask] = cv2.addWeighted(img, alpha, img, 1 - alpha, 0)[mask]  
  
 cv2.imshow("Image", out)  
 cv2.waitKey(1)

**RESULT**

**#DRAW SOLID:-**



**#DRAW TRANSPERENCY:-**



**CONCLUSION**

Demonstration of the image processing capabilities of OpenCV. The ultimate goal is to create a computer vision machine learning application that promotes Human computer interaction (HCI) also named ManMachine Interaction (MMI)] refers to the relation between the human and the computer or more precisely the machine, and since the machine is insignificant without suitable utilize by the human there are two main characteristics should be deemed when designing a HCI system as mentioned: functionality and usability. System functionality referred to the set of functions or services that the system equips to the users while system usability

**FUTURE SCOPE**

Some examples that are being implemented in the near future are as follows:

Video Processing using Android Phone:

Mobile devices such as smart phones, iPads and tablet pcs are equipped with cameras, the demand of the image processing applications increased. These applications need to be faster and consumes lower power because the mobile device is only powered by a battery. The hardware technology depends on the semiconductor technology instead we can use an efficient programming language to write an image processing application for the mobile devices.

Robot Control:

Controlling the robot using gestures considered as one of the interesting applications in this field proposed a system that uses the numbering to count the five fingers for controlling a robot using hand pose signs. The orders are given to the robot to perform a particular task, where each sign has a specific meaning and represents different function.

**Challenges Faced**

There were many challenges faced by me during this project. The very first issue I faced was of packages. First of all I installed latest version of pycharm and packages too so nothing goes wrongs, but it was opposite for this handdetection module to move object virtually we have to use old version like cvzone package(ver. 1.4.1) and Mediapipe package(ver. 0.8.7).I used latest version package so it was not detecting hands movements and objects. It, will take only videocapture. So, make sure before installing any packages first check his versions.

**REFERENCES**

* Volume 5, Issue 1, January 2015 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering : Research Paper -- Gesture Controlled Computer.
* https://towardsdatascience.com/Painting-withopencv/
* https://www.geeksforgeeks.org/live-webcamdrawing-using-opencv/
* OpenCV for Computer Vision Applications, Proceedings of National Conference on Big Data and Cloud Computing (NCBDC’15), March 2015
* Numpy.org, 2017. Online]. Available: http://www.numpy.org
* Study on Object Detection using Open CV – Python, International Journal of Computer Applications (0975 – 8887) Volume 162 – No 8, March 2017
* Nidhi, “Image Processing and Object Detection”, Dept. of Computer Applications, NIT, Kurukshetra, Haryana, 1(9): 396-399, 2015