SONOPANT DANDEKAR SHIKSHAN MANDALI'S

A

Project Report On

AI VIRTUAL MOUSE BY USING PYTHON

SUBMITTED BY

Mr.ROHIT RAOSAHEB PHATANGARE

66032

UNDER THE GUIDANCE OF

Dr.YUGANDHARA MORE

THIRD-YEAR BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

SEMESTER-VI

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 **ROHIT RAOSAHEB PHATANGARE** is a student studying in TY.BSC.CS SEM-VI. She has completed project work entitled **AI VIRTUAL MOUSE BY USING PYTHON** under the guidance of Faculty Member **Dr.YUGANDHARA MORE** satisfactorily and has submitted it to the University of Mumbai in partial fulfillment 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**

SmallSEOTools

PLAGIARISM REPORT

Words 1973
Characters 3355

Date: March 23,2022
Excluded URL

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

Content Checked For Plagiarism

Plagiarism checker software evaluates submitted documents for plagiarism against other databases of content. Writers and content creators leverage plagiarism checkers to avoid accidentally plagiarizing in their work, while organizations use these tools to ensure documents sent to them are free of plagiarized content. Many plagiarism checkers are designed for academia and utilized by educators and administrators to examine written work submitted to them by current and prospective students. Private businesses and organizations also employ these solutions for a multitude of uses, most often to check for plagiarism in content written by current and prospective employees.

Plagiarism checkers are often built into other proofreading tools so writers can monitor their work for plagiarism while avoiding grammar and spelling

errors. Some AI writing assistants will include plagiarism checking functionality with their offering in addition to other features.

To qualify for inclusion in the Plagiarism Checker category, a product must:-

Allow users to upload text-based documents and/or monitor input text in real-time
Highlight plagiarized content within a written body of work
Refer to one or more content repositories when searching for plagiarism

LIST OF ABBREVIATIONS

GUI = Graphical User Interface

LED = Light Emitting Diodes

HCI = Human Computer Interaction

MHI = Motion History Images

IDE = Integrated Development Environment

OpenCV = Open Source Computer Vision

HSV = Hue, Saturation, and Values

RGB = Red, Green, and Blue

INDEX

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 **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 **Dr.YUGANDHARA MORE** my internal project guide, who has been a constant source of motivation and inspiration and has helped in bringing out the best in me. I am also thankful to all staff members of the computer science department who as a team have contributed to the successful completion of this project.

Place: Palghar

Date:

Rohit Phatangare

Download and Installation Instructions for

Python for Windows

Let's Get Started!

Note: All highlights below are added for emphasis and are not present on the actual pages.

Downloading the Python Version 3..9.6 :

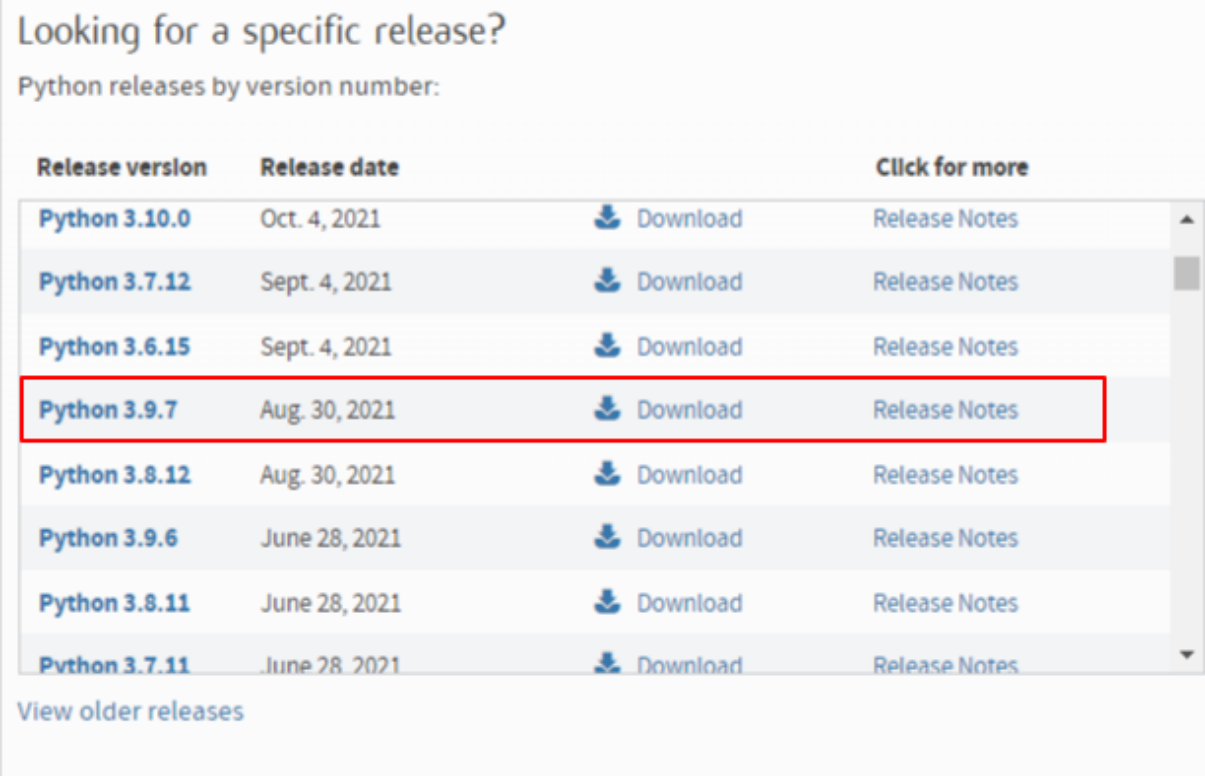
The Python download requires about 25 Mb of disk space; keep it on your machine, in case you need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space

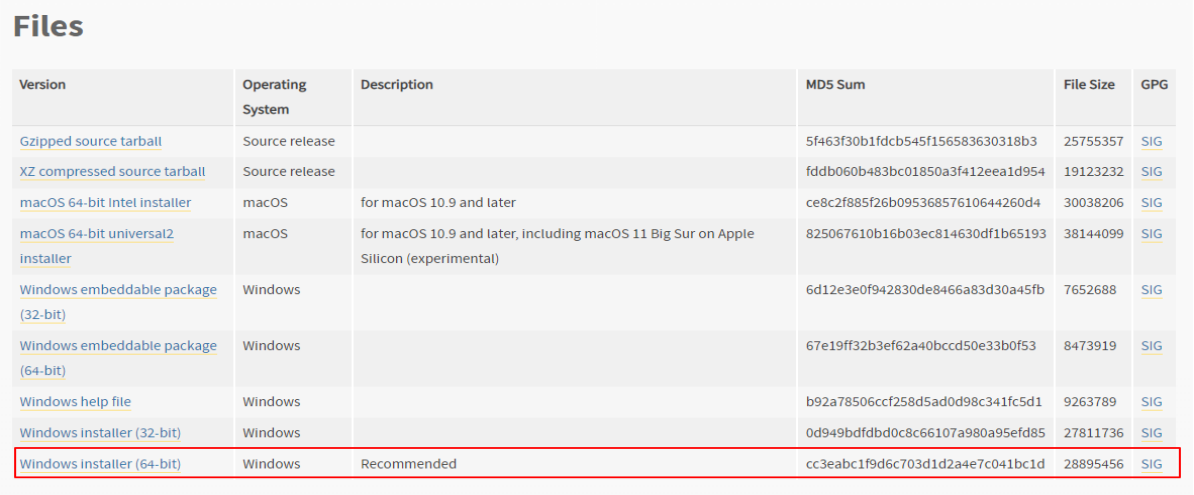
Downloading

1) Click Python Download.
The following page will appear in your browser.

2) Click the **Download Python 3.9.6** button. The following pop-up window titled **Opening python-3.9.6-amd64.exe** will appear.

- Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).
- Feel free to explore this webpage further; if you want to just continue the installation, you can terminate the tab browsing this webpage.
- Start the **Installing** instructions directly below.



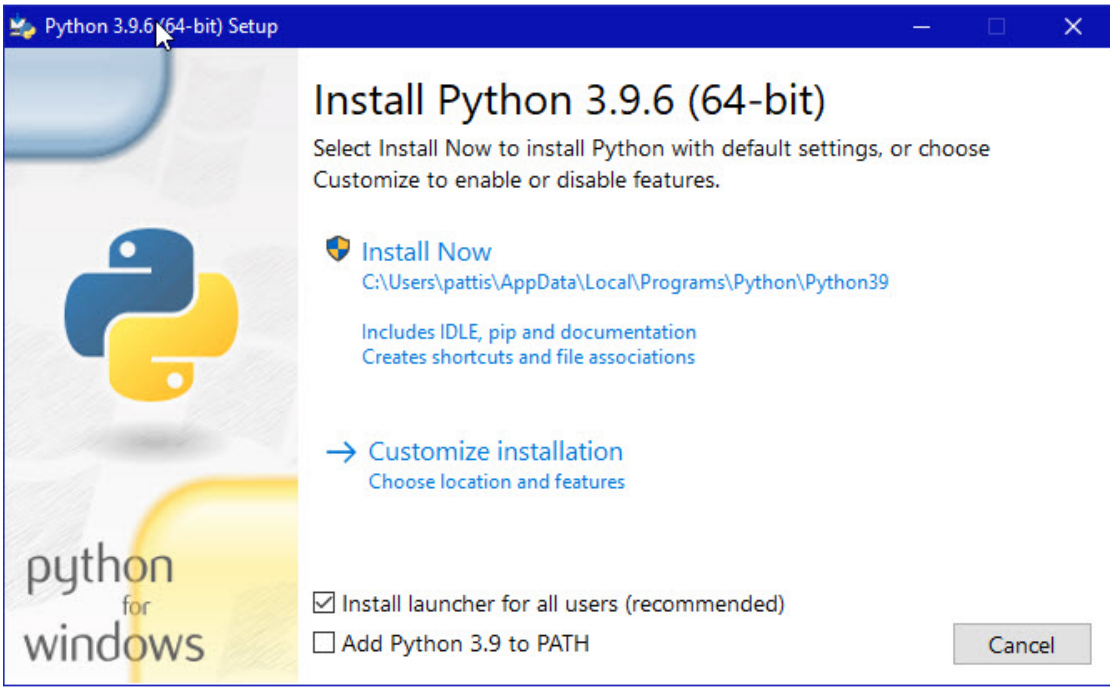


Files

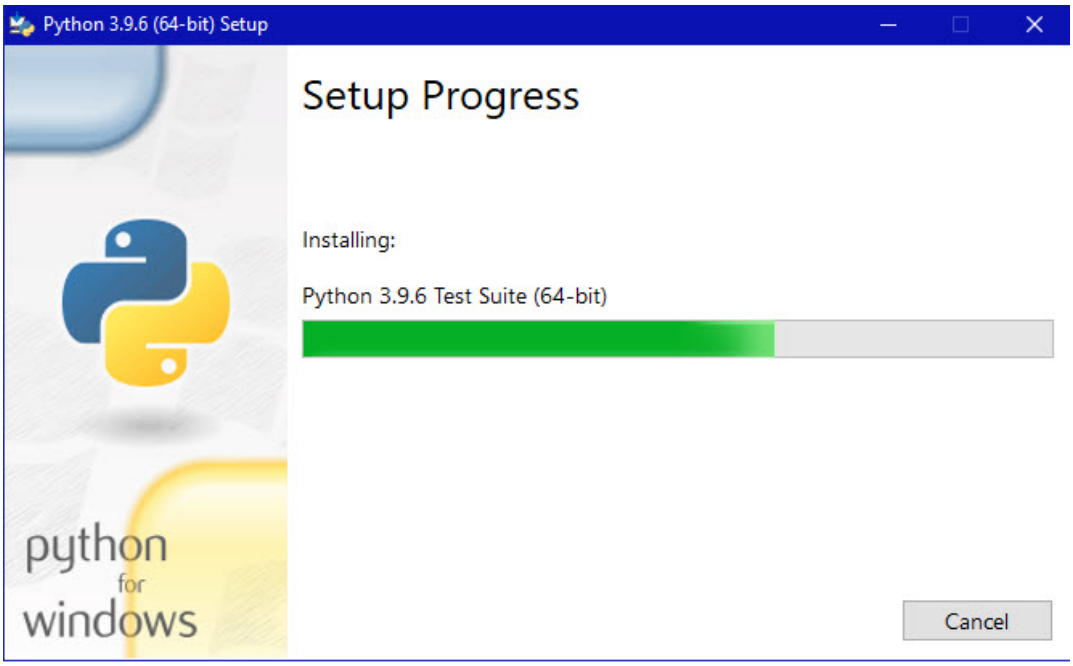
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		5f463f30b1fdcb545f156583630318b3	25755357	SIG
XZ compressed source tarball	Source release		fddb060b483bc01850a3f412eea1d954	19123232	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	ce8c2f885f26b09536857610644260d4	30038206	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	825067610b16b03ec814630df1b65193	38144099	SIG
Windows embeddable package (32-bit)	Windows		6d12e3e0f942830de8466a83d30a45fb	7652688	SIG
Windows embeddable package (64-bit)	Windows		67e19ff32b3ef62a40bccd50e33b0f53	8473919	SIG
Windows help file	Windows		b92a78506ccf258d5ad0d98c341fc5d1	9263789	SIG
Windows installer (32-bit)	Windows		0d949bdfdbd0c8c66107a980a95efd85	27811736	SIG
Windows installer (64-bit)	Windows	Recommended	cc3eabc1f9d6c703d1d2a4e7c041bc1d	28895456	SIG

Installing

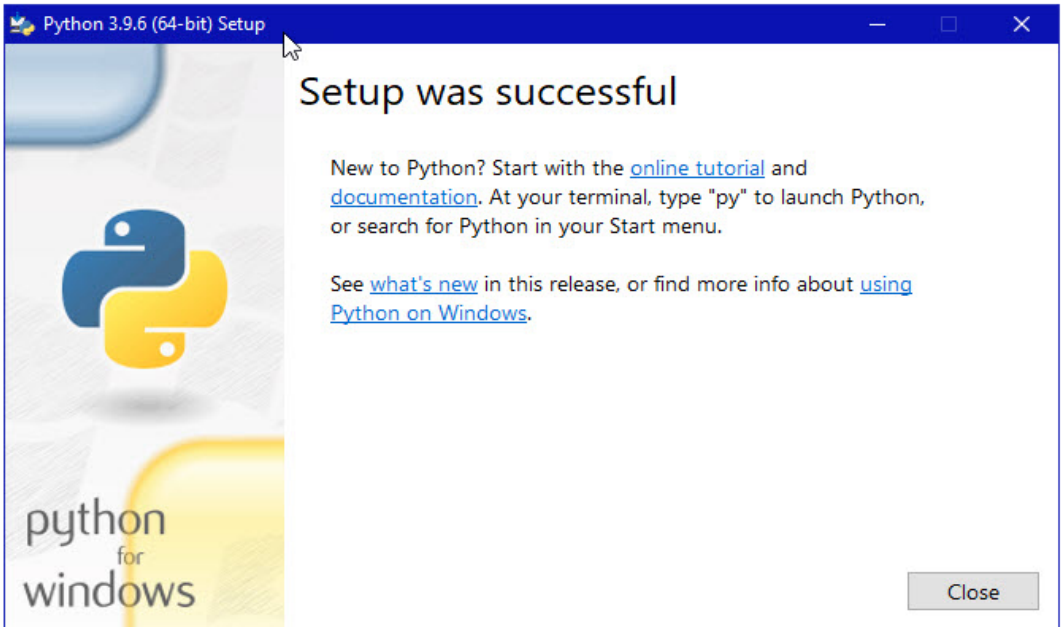
1. Double-click the icon labeling the file **python-3.9.7-amd64.exe**.
   A **Python 3.9.7 (64-bit) Setup** pop-up window will appear.



Click the **Install Now** button.
A new **Python 3.9.6 (64-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar.



During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.9.6 (64-bit) Setup** pop-up window will appear with a **Setup was successfuly** message.



Click the **Close** button.

Python should now be installed.

Installation Of PyCharm IDE

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects:

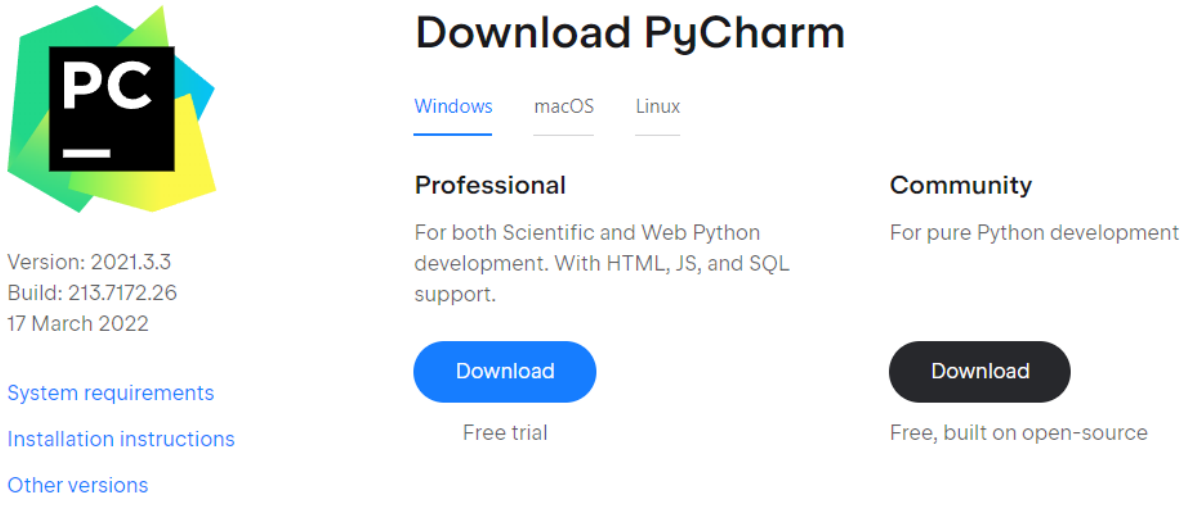
- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

Features of PyCharm Besides : a developer will find PyCharm comfortable to work with because of the features mentioned below

Steps Involved :

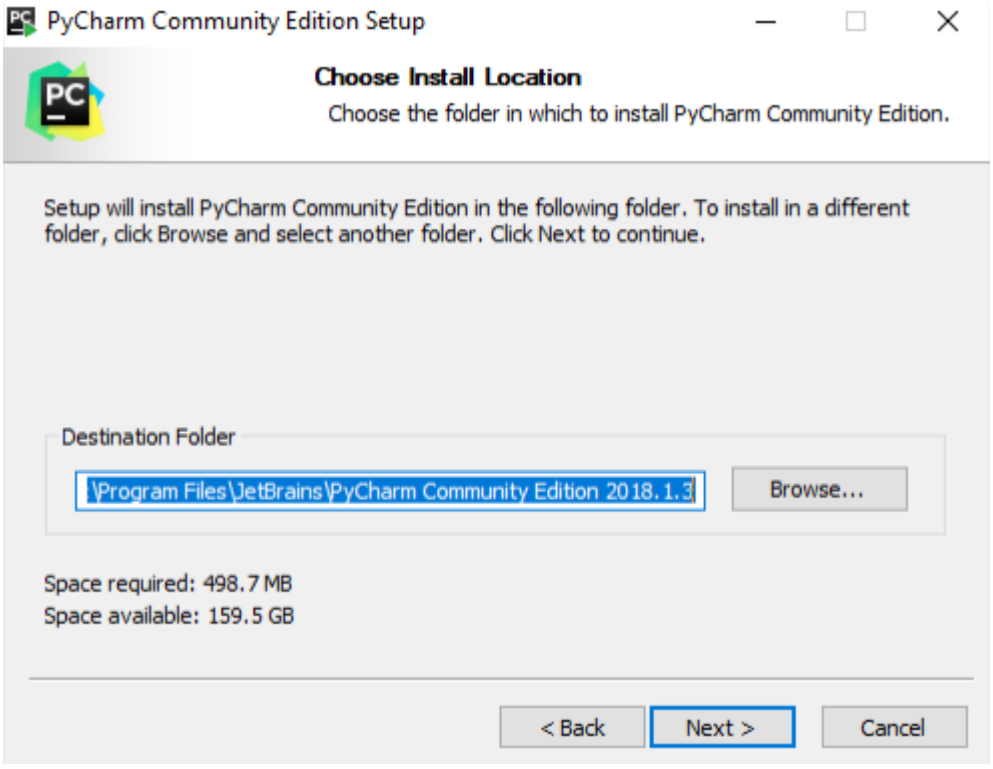
You will have to follow the steps given below to install PyCharm on your system. These steps show the installation procedure starting from downloading the PyCharm package from its official website to creating a new project.

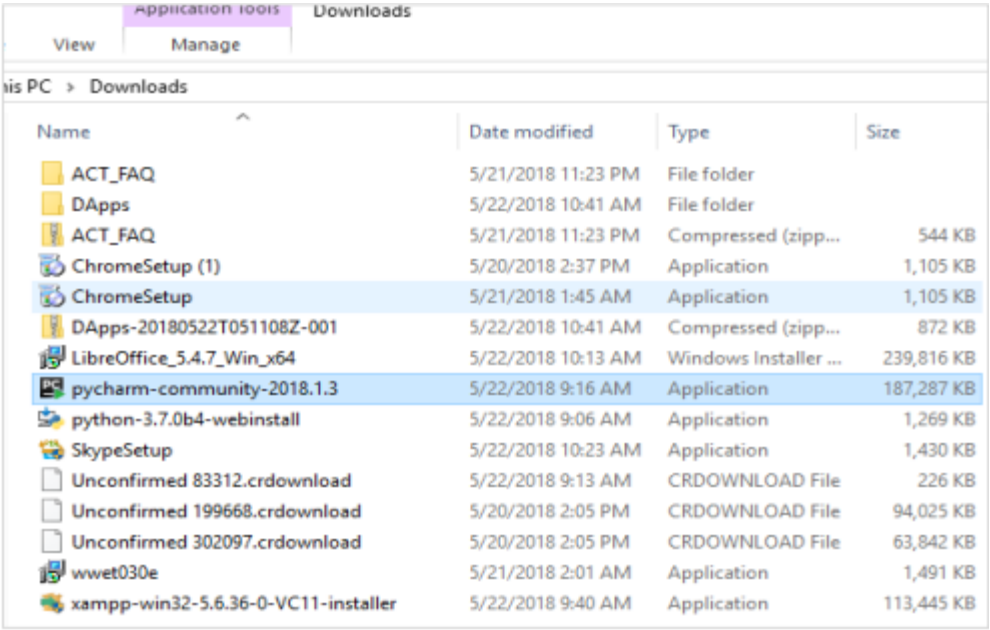
Step 1 Download the required package or executable from the official website PyCharm https://www.jetbrains.com/pycharm/download/#section=windows. Here you will observe two versions of package for Windows as shown in the screenshot given below:



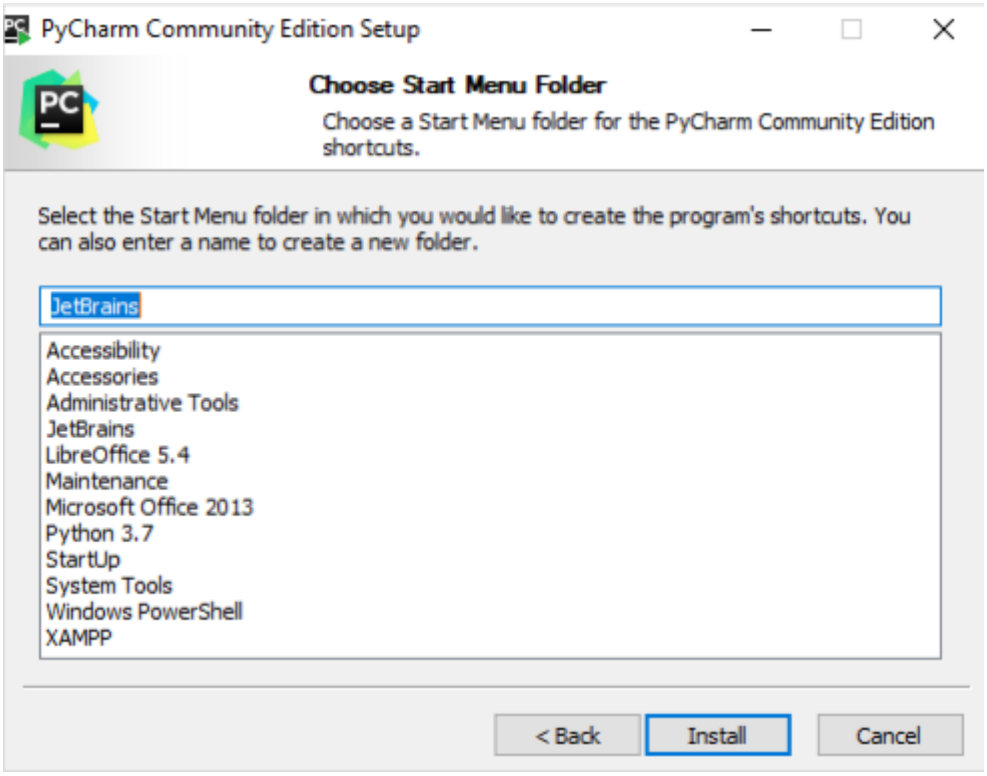
Note that the professional package involves all the advanced features and comes with free trial for few days , Note that we will continue with **community package** throughout this tutorial.

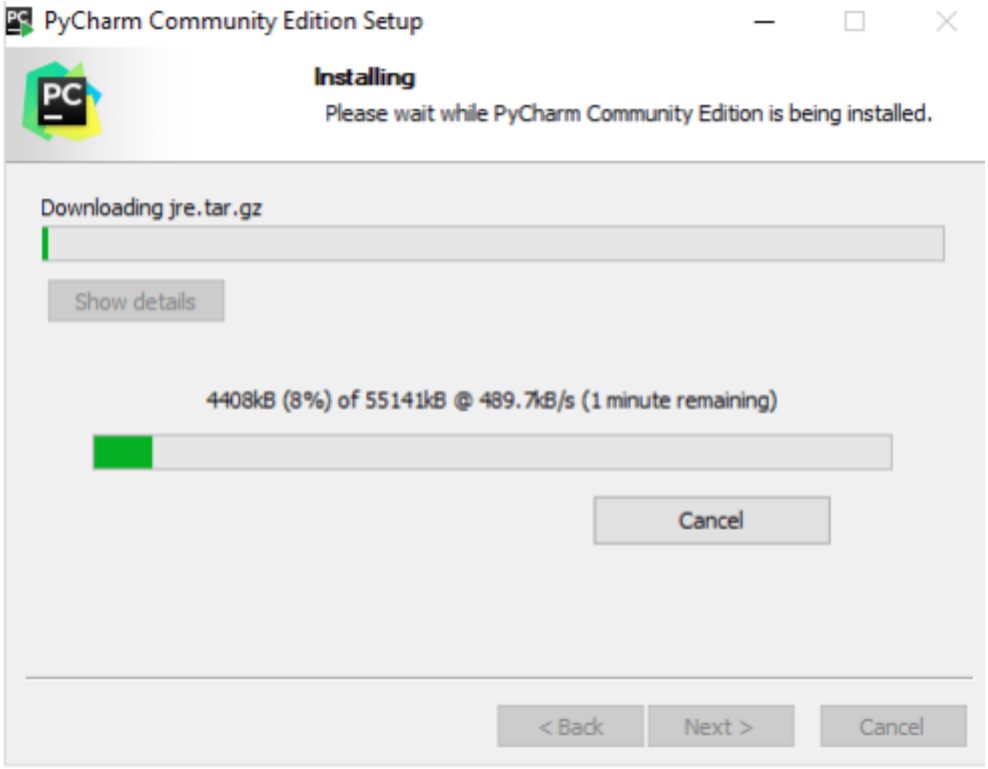
Step 2 : Download the community package (executable file) onto your system and mention a destination folder as shown below



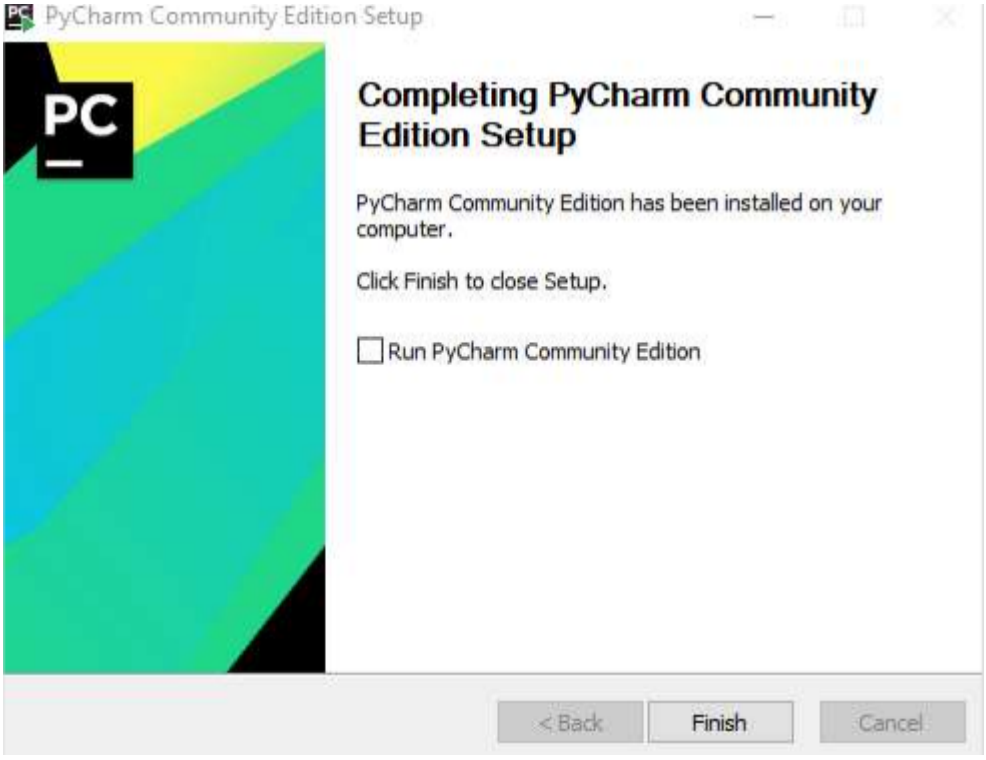


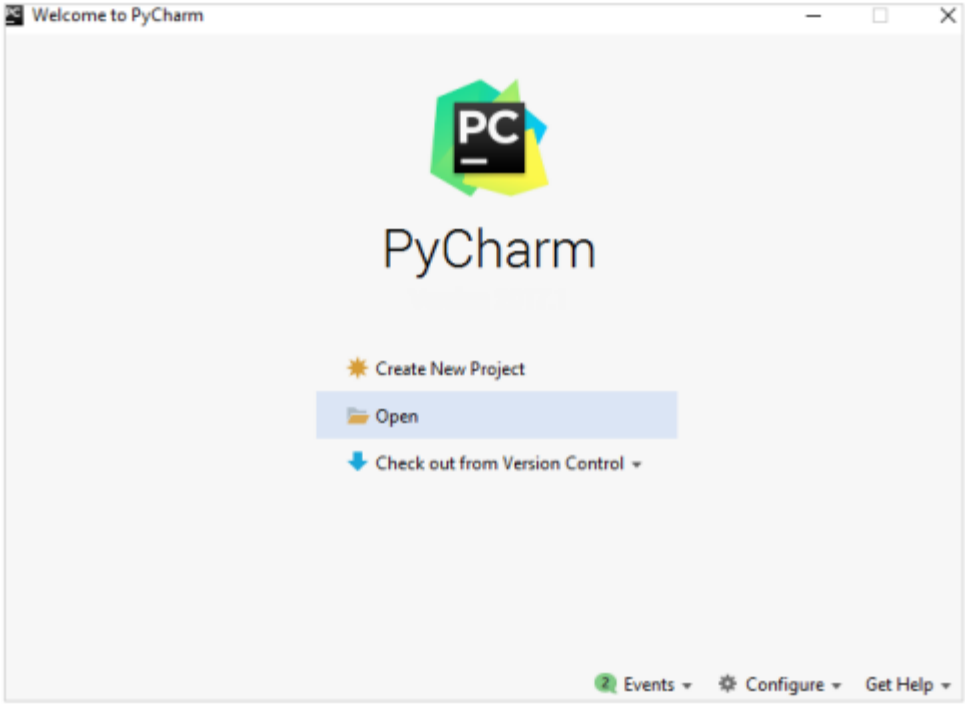
Step 3 Now, begin the installation procedure similar to any other software package.





Step 4 Once the installation is successful, PyCharm asks you to import settings of the existing package if any.





Topic: **AI VIRTUAL MOUSE BY USING PYTHON**

Name: Rohit Raosaheb Phatangare . **Roll No**.= 66032

1.INTRODUCTION :

- A mouse, in computing terms is a pointing device that detects two-dimensional movements relative to a surface.
- This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform.
- There are a lot of different types of mouse that have already existed in the modern days technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects table top movement and then sends off the information to the computer for processing. On the year 2004, the laser mouse was then introduced to improve the accuracy movement with the slightest hand movement, it overcome the limitations of the optical mouse which is the difficulties to track high-gloss surfaces.
- Despite the limitations, the computer technology still continues to grow, so does the importance of the human computer interactions. Ever since the introduction of a mobile device that can be interact with touch screen technology, the world is starting to demand the same technology to be applied on every technological devices, this includes the desktop system. However, even though the touch screen technology for the desktop system is already exist, the price can be very steep.
- Therefore, a virtual human computer interaction device that replaces the physical mouse or keyboard by using a webcam or any other image capturing devices can be an alternative way for the touch screen.
- This device which is the webcam will be constantly utilized by a software that monitors the gestures given by the user in order to process it and translate to motion of a pointes, as similar to a physical mouse.

1.2. Review Of The Physical Mouse

It is known that there are various types of physical computer mouse in the modern technology, the following will discuss about the types and differences about the physical mouse.

1.2.1. Mechanical Mouse

- Known as the trackball mouse that is commonly used in the 1990s, the ball within the mouse are supported by two rotating rollers in order to detect the movement made by the ball itself. One roller detects the forward/backward motion while the other detects the left/right motion.
- The common functions included are the left/right buttons and a scroll-wheel. However, due to the constant friction made between the mouse ball and the rollers itself.

Advantage	Disadvantage
Allows the users to control the computer system by moving the mouse	Prone to degradation of the mouse rollers and button switches, causing to be faulty
Provides precise mouse tracking movements	Requires a flat surface to operate

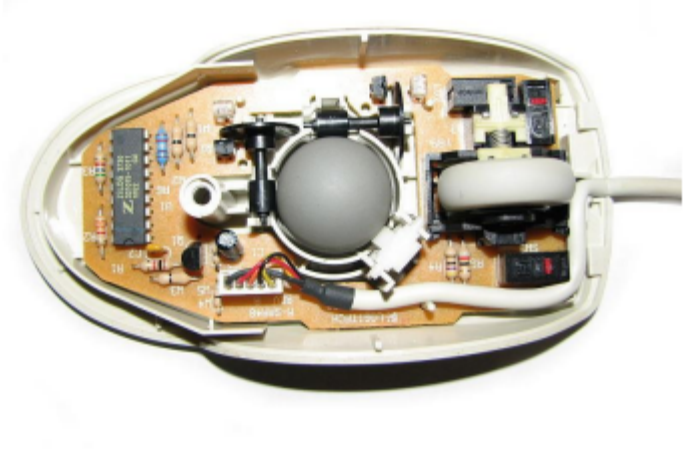


Figure 1.1 Mechanical mouse, with top cover removed

1.2.2. Optical And Laser Mouse

- A mouse that commonly used in these days, the motions of optical mouse rely on the Light Emitting Diodes (LEDs) to detect movements relative to the underlying surface, while the laser mouse is an optical mouse that uses coherent laser lights.
- long term usage without a proper cleaning or maintenance may leads to dust particles trap between the LEDs, which will cause both optical and laser mouse having surface detection difficulties.
- Other than that, it's still prone to degradation of the button switches, which again will cause the mouse to function improperly unless it was disassembled and repaired.

Advantage	Disadvantage
Allows better precision with lesser hand movements.	Prone to button switches degradation
Longer life-span.	Does not function properly while on a polished surface.

Figure 1.2 Optical Mouse, with top cover removed

1.3. Problem Statement

It's no surprised that every technological devices have its own limitations, especially when it comes to computer devices.

- Physical mouse is subjected to mechanical wear and tear.
- Physical mouse requires special hardware and surface to operate.
- Physical mouse is not easily adaptable to different environments and its performance varies depending on the environment.
- Mouse has limited functions even in present operational environments. All wired mouse and wireless mouse have its own lifespan.

1.4. Motivation of Virtual Mouse :

It is fair to say that the Virtual Mouse will soon to be substituting the traditional physical mouse in the near future, as people are aiming towards the lifestyle where that every technological devices can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provides convenience, but it's cost effective as well

1.4.1. Convenient :

It is known in order to interact with the computer system, users are required to use an actual physical mouse, which also requires a certain area of surface to operate, not to mention that it suffers from cable length limitations. Virtual Mouse requires none of it, as it only a webcam to allow image capturing of user's hand position in order to determine the position of the pointers that the user want it to be

1.4.2. Cost Effective :

A quality physical mouse is normally cost from the range of 30 ringgit to a hefty 400 ringgit, depending on their functionality and features. Since the Virtual Mouse requires only a webcam, a physical mouse are no longer required, thus eliminating the need to purchase one, as a single webcam is sufficient enough to allow users to interact with the computer system through it, while some other portable computer system such as the laptop, are already supplied with a built-in webcam, could simply utilize the Virtual Mouse software without having any concerns about purchasing any external peripheral devices.

1.5. Project Scope :

Virtual Mouse that will soon to be introduced to replace the physical computer mouse to promote convenience while still able to accurately interact and control the computer system

Therefore, this project will develop a software application with the aid of the latest software coding technique and the open-source computer vision library also known as the OpenCV. The scope of the project is as below:

Real time application. User friendly application. Removes the requirement of having a physical mouse.

1.6. Impact, Significance and Contribution:

The Virtual Mouse application is expected to replace the current methods of utilizing a physical computer mouse where the mouse inputs and positions are done manually. This application offers a more effortless way to interact with the computer system, where every task can be done by gestures

2.0. Mouse Simulation Using Two Coloured Tapes :

Kamran Niyazi et al. (2012), mentioned that to solve the stated problem, ubiquitous computing method is required. Thus, colour tracking mouse simulation was proposed. The said system tracks two colour tapes on the user fingers by utilizing the computer vision technology

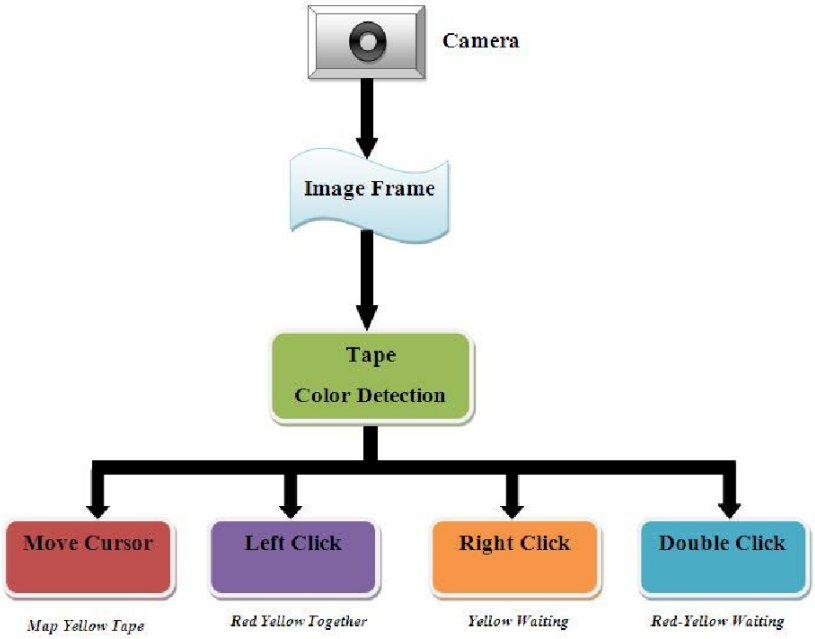


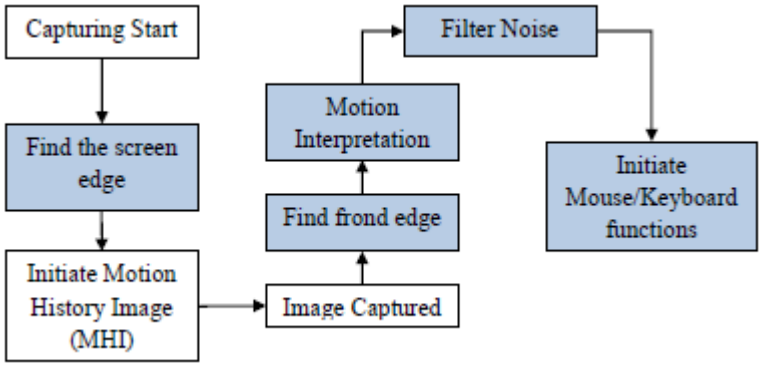
Figure 1: System Architecture

2.1. Virtual Mouse Using a Webcam

Another colour detection method proposed by Kazim Sekeroglu (2010),The proposed system are capable of detecting the pointers by referring the defined colour information, track the motion of the pointers, move the cursor according to the position of the pointer, and simulate the single and double left or/and right click event of the mouse.

2.2. Portable Vision-Based Human Computer Interaction(HCI)

Another "Ubiquitous Computing" approach proposed by Chu-Feng Lien (2015), requires only finger-tips to control the mouse cursor and click events. The proposed system doesn't requires hand-gestures nor colour tracking in order to interact with the system, instead it utilize a feature name Motion History Images(MHI) , a method that used to identify movements with a row of images in time.



2.3. Virtual Mouse Using a Webcam :

Another colour detection method proposed by Kazim Sekeroglu (2010), the system requires three fingers with three colour pointers to simulate the click events. The proposed system are capable of detecting the pointers by referring the defined colour information, track the motion of the pointers, move the cursor according to the position of the pointer, and simulate the single and double left or/and right click event of the mouse.

3.0. Recognition Phase :

a) Webcam & Variables Initialization On the early stage of the recognition phase, the program will initialize the required variables which will be used to hold different types of frames and values where each are will be used to carry out certain task. Furthermore, this is the part where the program collects the calibrated HSV values and settings where it will be used later during the transitions of Binary Threshold

b) HSV Frame Transition : The captured frame require to be converted from a BGR format to a HSV format. Which can be done by using cvtColor(src, dst, CV_BGR2HSV).

c) Real Time Image Acquisition: The real time image is captured by using the webcam by using (cv::VideoCapture cap(0);), where every image captured are stored into a frame variable (cv::Mat), which will be flipped and compressed to a reasonable size to reduce process load.

d) Colour Combination Comparison After obtaining results from Morphological Transformation process, the program will calculate the remaining number of objects by highlighting it as blobs, this process requires cvBlob library, which is an add-on to OpenCV. The results of the calculation will then send for comparison to determine the mouse functions based on the colour combinations found within the captured frames.

★ Execution of Mouse Action :

The program will executes the mouse actions based on the colours combinations exist in the processed frame. The mouse actions will perform according to the coordinates provided by the program, and the program will continue on acquire and process the next real-time image until the users exit from the program.

3.1. Implementation Issues and Challenges :

Throughout the development of the application, there are several implementation issues occurred. The following describes the issues and challenges that will likely to be encountered throughout the development phase:

- **The interruptions of salt and pepper noises within the captured frames** :

Salt and pepper noises occurred when the captured frame contains required HSV values that are too small, but still underwent a series of process even though it's not large enough to be considered an input. To overcome this issue, the unwanted HSV pixels within the frame must first be filtered off, this includes the area of the pixels that are too large and small. With this method, the likelihood of interruptions of similar pixels will reduce greatly.

- **Performance degradation due to high process load for low-tier system :**

Since the application is required to undergo several of process to filter, process and execute the mouse functions in real time, the application can be CPU intensive for most of the low-tier system. If the size of the captured frames is too large, the time-taken for the application to process the entire frame are increase drastically. Therefore, to overcome this issue, the application is required to process only the essential part of the frames, and reduces the redundant filtering process that could potentially slow the application down.

The difficulties of calibrating the brightness and the contrast of the frames to get the required HSV values. :

The intensity of brightness and contrast matters greatly when it comes to acquiring the required colour pixels To overcome this issue, the application must first start up with a calibration phase, which allows the users to choose their desired colour pixels before directing them to the main phase.

4.0. Methodology :-

For this project we'll be using the Agile Software Development methodology approach in developing the application. The stated approach is an alternative to the traditional waterfall model that helps the project team respond to unpredictability through incremental and iterative work . The following describes the principles of the Agile Software Development methodology:

- Satisfy the customer by early and continuous delivery of workable software
- Encourage changes of requirement.
- Workable software is delivered frequently.
- Continuous collaboration between the stakeholders and the developers.
- Project are developed around motivated individuals. Encourage informal meetings.
- Operational software is the principle measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams Regular adaption to changing circumstances

<u>The following describes the phases within the agile methodology approach:</u>

- **<u>Planning</u>**

A thorough planning will be conducted in this phase where the existing systems/product, for this case, physical computer mouse will be reviewed and studied to identify the problems existed, a comparison of problems will be made to compare which problems are more crucial and requires improvement. An outline objective and the scope will be identified in order to provide an alternative solution to the problem.

- **Requirement Analysis**

The phase that gathers and interpreting the facts, diagnosing problems and recommending improvements to the system. In this phase, the collected problem statements will be extensively studied in order to find a proper solution or at least an improvements to the proposed system. All proposed solutions will be converted into requirements where it will be documented in a requirement specification

- **Designing**

The requirement specification from the previous phase will be studied and prioritize to determine which requirement are more important where the requirement with the highest priority will be delivered first. After the study, the system design will be prepared as it helps in defining the overall system architecture and specifying the hardware and the software requirements.

- **Building**

The phase where the actual coding implementation takes place. By referring to the inputs from the system design, the system will be developed based on the prioritize requirements. However, due to we're using the agile methodology approach, the developed system will be considered as a prototype system where it will be integrated and tested by the users.

- **Testing**

The phase where the prototype system going through a series of test. The prototype system will first undergo integration where the features from the previous iteration cycle are added to the latest cycle. If the users requires additional features or modification, feedback gathering will be conducted, which resulted in further modification of the requirements and features where it will recorded and documented for the requirement analysis phase on the next iteration.

4.1. Hardware Requirement

Computer Desktop or Laptop The computer desktop or a laptop will be utilized to run the visual software in order to display what webcam had captured. A notebook which is a small, lightweight and inexpensive laptop computer is proposed to increase mobility.

System will be using :

Processor : Core2Duo

Main Memory : 4GB RAM

Hard Disk : 320GB Display : 14" Monitor

- Webcam Webcam is utilized for image processing, the webcam will continuously taking image in order for the program to process the image and find pixel position.

4.2. Software Requirement

The following describes the software needed in-order to develop the Virtual Mouse application:

Python Language :

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.Python is a programming language that lets you work quickly and integrate systems more efficiently.

OpenCV :

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

Software will be using:

OS : Window 10 Ultimate 64-bit

Language : Python , **Tool Used** : Open CV

CONCLUSION

In conclusion, it's no surprised that the physical mouse will be replaced by a virtual non-physical mouse in the Human-Computer Interactions (HCI), where every mouse movements can be executed with a swift of your fingers everywhere and anytime without any environmental restrictions. This project had develop a colour recognition program with the purpose of replacing the generic physical mouse without sacrificing the accuracy and efficiency, it is able to recognize colour movements, combinations, and translate them into actual mouse functions. Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented.

First and foremost, the coordinates of the colours that are in charge of handling the cursor movements are averaged based on a collections of coordinates, the purpose of this technique is to reduce and stabilize the sensitivity of cursor movements, as slight movement might lead to unwanted cursor movements

The purpose of this implementation is to promote convenience in controlling the program without much of a hassle. Therefore, actual mouse functions can be triggered accurately with minimum trial and errors.

5.0. Future Works :-

There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments

The following describes the improvements and the features required:

a) Smart Recognition Algorithm

Due to the current recognition process are limited within 25cm radius, an adaptive zoom-in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam.

b) Better Performance

The response time are heavily rely on the hardware of the machine, this includes the processing speed of the processor, the size of the available RAM, and the available features of webcam. Therefore, the program may have better performance when it's running on a decent machines with a webcam that performs better in different types of lightings.

REFERENCES

- https://www.geeksforgeeks.org/opencv-overview/
- https://www.javatpoint.com/opencv
- https://stackoverflow.com/questions/69594197/python-cvzone-why-do-i-get-this-valueerror
- https://github.com/explore
- https://www.youtube.com/watch?v=oXlwWbU8l2o&ab_channel=freeCodeCamp.org
- https://www.amazon.in/Mastering-OpenCV-Python-practical-processing/dp/1789344913/ref=asc_df_1789344913/?tag=googleshopdes-21&linkCode=df0&hvadid=397009686187&hvpos=&hvnetw=g&hvrand=23707000346472045&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9300465&hvtargid=pla-683868272770&psc=1&ext_vrnc=hi

CODE :-

AiVirtualMouseProject.py :-

```
import cv2
import numpy as np
import HandTrackingModule as htm
import time
import autopy

########################
wCam, hCam = 640, 480
frameR = 100     #Frame Reduction
smoothening = 7  #random value
########################

pTime = 0
plocX, plocY = 0, 0
clocX, clocY = 0, 0

cap = cv2.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)

detector = htm.handDetector(maxHands=1)
wScr, hScr = autopy.screen.size()

# print(wScr, hScr)

while True:
    # Step1: Find the landmarks
    success, img = cap.read()
    img = detector.findHands(img)
    lmList, bbox = detector.findPosition(img)

    # Step2: Get the tip of the index and middle finger
    if len(lmList) != 0:
        x1, y1 = lmList[8][1:]
        x2, y2 = lmList[12][1:]

        # Step3: Check which fingers are up
        fingers = detector.fingersUp()
        cv2.rectangle(img, (frameR, frameR), (wCam - frameR, hCam - frameR),
                      (255, 0, 255), 2)
```

```
        # Step4: Only Index Finger: Moving Mode
        if fingers[1] == 1 and fingers[2] == 0:

            # Step5: Convert the coordinates
            x3 = np.interp(x1, (frameR, wCam-frameR), (0, wScr))
            y3 = np.interp(y1, (frameR, hCam-frameR), (0, hScr))

            # Step6: Smooth Values
            clocX = plocX + (x3 - plocX) / smoothening
            clocY = plocY + (y3 - plocY) / smoothening

            # Step7: Move Mouse
            autopy.mouse.move(wScr - clocX, clocY)
            cv2.circle(img, (x1, y1), 15, (255, 0, 255), cv2.FILLED)
            plocX, plocY = clocX, clocY

        # Step8: Both Index and middle are up: Clicking Mode
        if fingers[1] == 1 and fingers[2] == 1:

            # Step9: Find distance between fingers
            length, img, lineInfo = detector.findDistance(8, 12, img)

            # Step10: Click mouse if distance short
            if length < 40:
                cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0), cv2.FILLED)
                autopy.mouse.click()

    # Step11: Frame rate
    cTime = time.time()
    fps = 1/(cTime-pTime)
    pTime = cTime
    cv2.putText(img, str(int(fps)), (28, 58), cv2.FONT_HERSHEY_PLAIN, 3,
(255, 8, 8), 3)

    # Step12: Display
    cv2.imshow("Image", img)
    cv2.waitKey(1)
```

HandTrackingModule.py :-

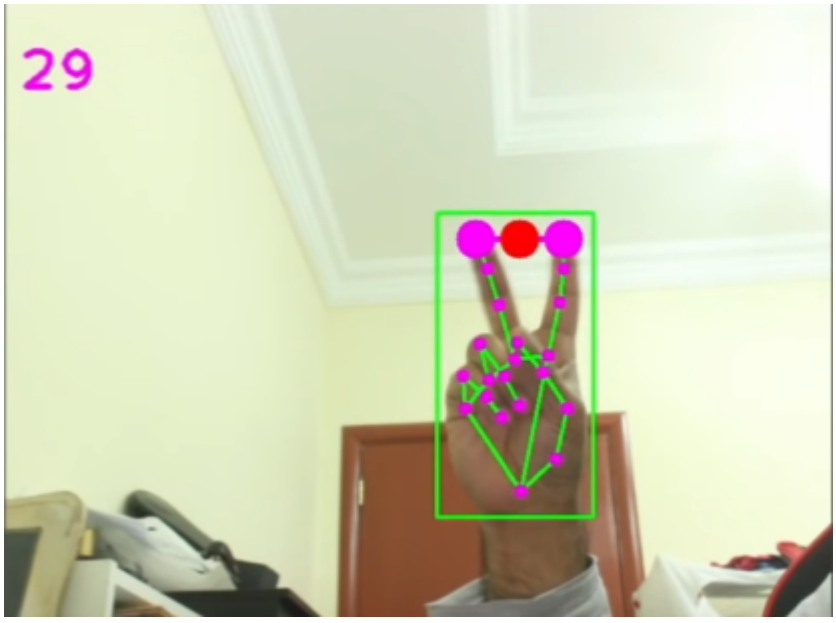
```
import cv2
```

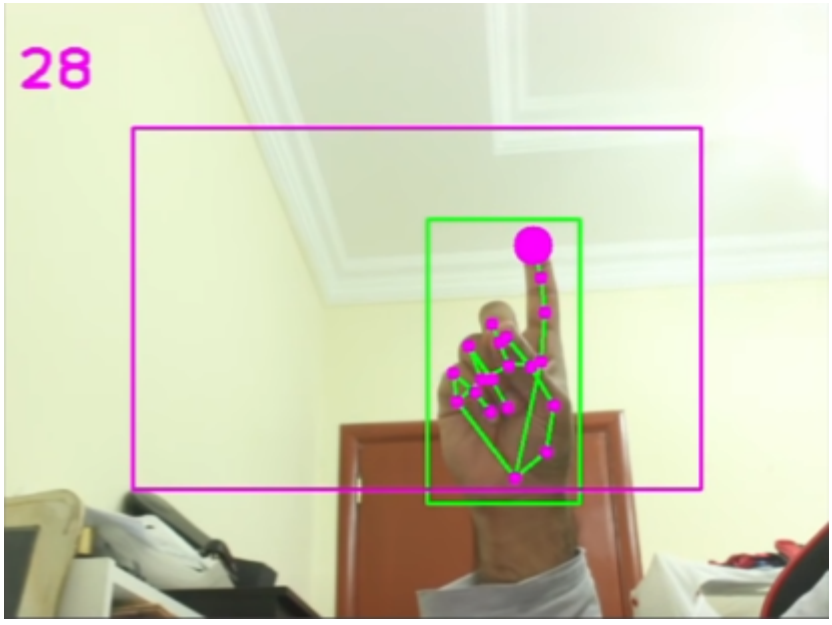
```
import mediapipe as mp
import time
import math
import numpy as np


class handDetector():
    def __init__(self, mode=False, maxHands=2, detectionCon=0.5,
trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils
        self.tipIds = [4, 8, 12, 16, 20]

    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
        # print(results.multi_hand_landmarks)

        if self.results.multi_hand_landmarks:
            for handLms in self.results.multi_hand_landmarks:
                if draw:
                    self.mpDraw.draw_landmarks(img, handLms,
self.mpHands.HAND_CONNECTIONS)
        return img

    def findPosition(self, img, handNo=0, draw=True):
        xList = []
        yList = []
        bbox = []
        self.lmList = []
        if self.results.multi_hand_landmarks:
            myHand = self.results.multi_hand_landmarks[handNo]
            for id, lm in enumerate(myHand.landmark):
            # print(id, lm)
                h, w, c = img.shape
```

```
                cx, cy = int(lm.x * w), int(lm.y * h)
                xList.append(cx)
                yList.append(cy)
                # print(id, cx, cy)
                self.lmList.append([id, cx, cy])
                if draw:
                    cv2.circle(img, (cx, cy), 5, (255, 0, 255), cv2.FILLED)

            xmin, xmax = min(xList), max(xList)
            ymin, ymax = min(yList), max(yList)
            bbox = xmin, ymin, xmax, ymax

            if draw:
                cv2.rectangle(img, (xmin - 20, ymin - 20), (xmax + 20, ymax + 20),
(0, 255, 0), 2)

        return self.lmList, bbox

    def fingersUp(self):
        fingers = []
    # Thumb
        if self.lmList[self.tipIds[0]][1] > self.lmList[self.tipIds[0] -1][1]:
            fingers.append(1)
        else:
            fingers.append(0)
    # Fingers
        for id in range(1, 5):
            if self.lmList[self.tipIds[id]][2] < self.lmList[self.tipIds[id] -2][2]:
                fingers.append(1)
            else:
                fingers.append(0)
    # totalFingers = fingers.count(1)
        return fingers

    def findDistance(self, p1, p2, img, draw=True, r=15, t=3):
        x1, y1 = self.lmList[p1][1:]
        x2, y2 = self.lmList[p2][1:]
        cx, cy = (x1 + x2) // 2, (y1 + y2) // 2

        if draw:
            cv2.line(img, (x1, y1), (x2, y2), (255, 0, 255), t)
            cv2.circle(img, (x1, y1), r, (255, 0, 255), cv2.FILLED)
```

```
            cv2.circle(img, (x2, y2), r, (255, 0, 255), cv2.FILLED)
            cv2.circle(img, (cx, cy), r, (0, 0, 255), cv2.FILLED)
        length = math.hypot(x2 - x1, y2 - y1)

        return length, img, [x1, y1, x2, y2, cx, cy]

def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(1)
    detector = handDetector()
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList, bbox = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime
        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)
        cv2.imshow("Image", img)
        cv2.waitKey(1)

if __name__ == "__main__":
    main()
```

Output :


29


28