DRONACHARYA
College of Engineering

PROJECT

Department of CSE(AIML)

AI Virtual Mouse Project

NAME: Gaurav Haritas & Vedika Yadav
BRANCH: CSE(AI & ML)
SEM: 4th
ROLL NO: 23253 & 23249

UNIV. REG. NO:

Algorithm Used for Hand Tracking

PYTHON: -

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

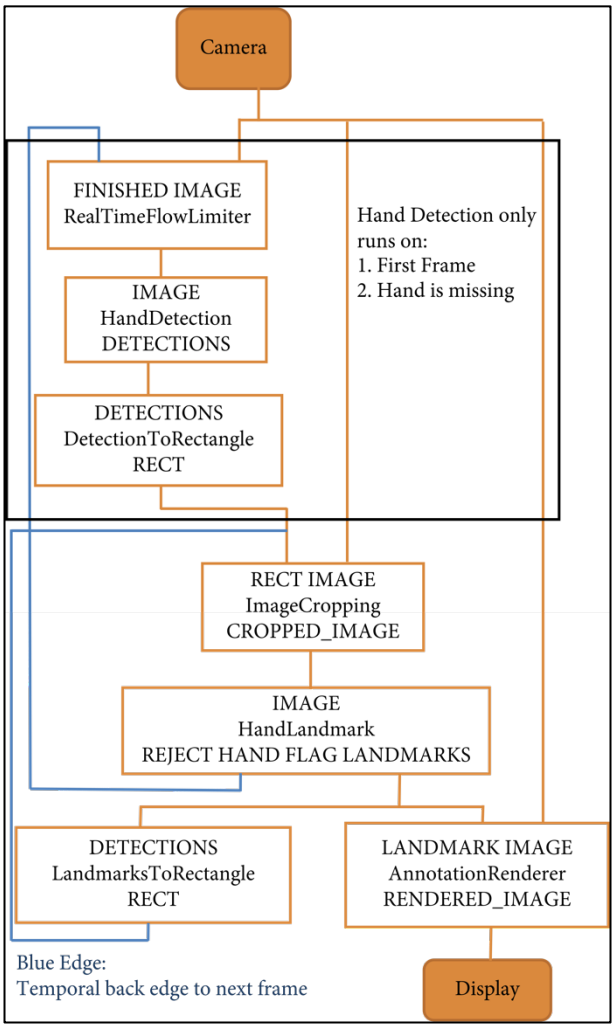
Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Algorithm Used for Hand Tracking

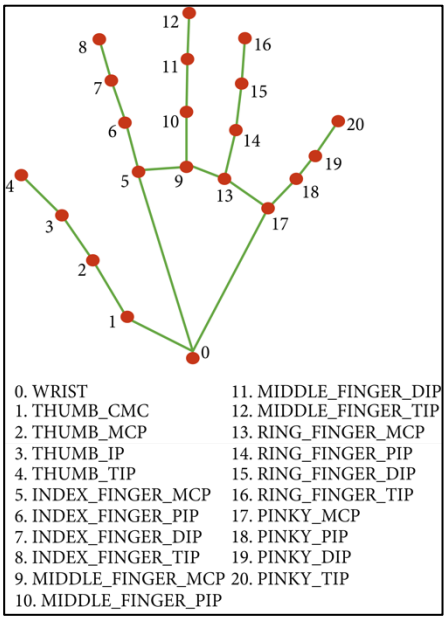
For the purpose of detection of hand gestures and hand tracking, the MediaPipe framework is used, and OpenCV library is used for computer vision. The algorithm makes use of the machine learning concepts to track and recognize the hand gestures and hand tip.

Mediapipe

MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos. The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose. The steps involved in the system that uses MediaPipe are carried out in the pipeline configuration. The pipeline created can run in various platforms allowing scalability in mobile and desktops. The MediaPipe framework is based on three fundamental parts; they are performance evaluation, framework for retrieving sensor data, and a collection of components which are called calculators, and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers are able to replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph is created with MediaPipe where each node is a calculator and the nodes are connected by streams.



Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the nonmaximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating 21 joint or knuckle co-ordinates in the hand region, as shown below.

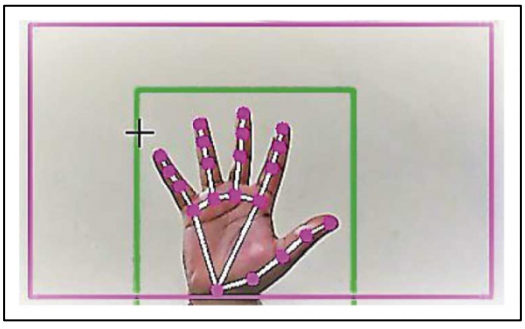


OpenCV

OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. The OpenCV library is used in image and video processing and also analysis such as face detection and object detection.

The Camera Used In The AI Virtual Mouse

The proposed AI virtual mouse system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library OpenCV, the video capture object is created and the web camera will start capturing video, as shown in Figure 4. The web camera captures and passes the frames to the AI virtual system.



Capturing The Video Processing

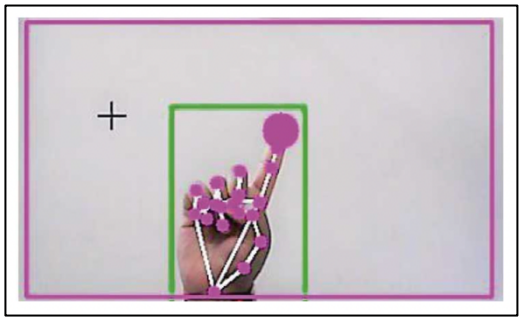
The AI virtual mouse system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB color space to find the hands in the video frame by frame as shown in the following code:

```
while video.isOpened():
    _,frame=video.read()
    image=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
    image=cv2.flip(image,1)
    imageHeight,imageWidth,_=image.shape
    results=hands.process(image)

    image=cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
```

Detecting Which Finger Is Up and Performing The Particular Mouse Function

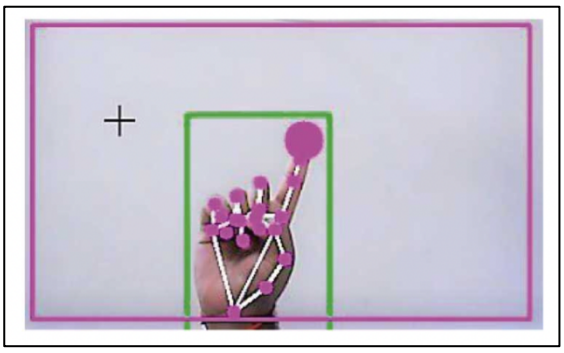
In this stage, we are detecting which finger is up using the tip Id of the respective finger that we found using the MediaPipe and the respective co-ordinates of the fingers that are up, as shown, and according to that, the particular mouse function is performed.



Mouse Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision

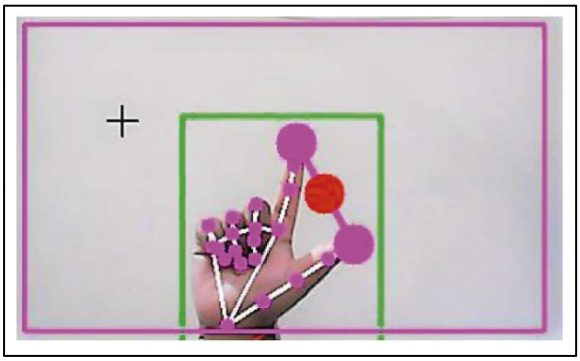
1. For the mouse cursor moving around the computer window.

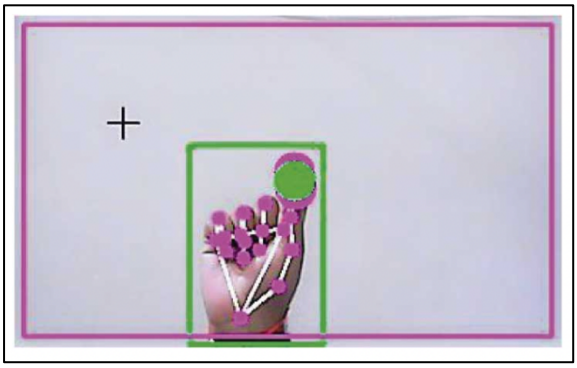
If the index finger is up with tip Id = 1 or both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up, the mouse cursor is made to move around the window of the computer using the AutoPy package of Python, as shown.



2. For the Mouse to Perform Left Button Click.

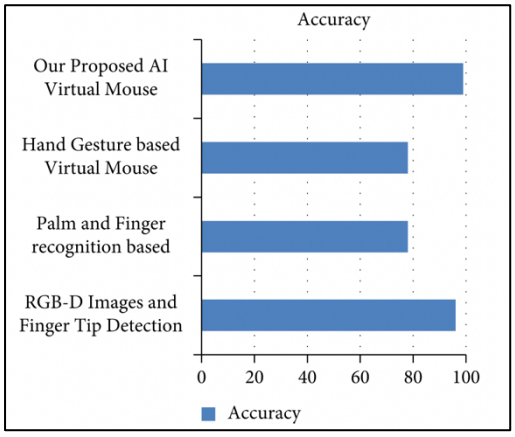
If both the index finger with tip Id = 1 and the thumb finger with tip Id = 0 are up and the distance between the two fingers is lesser than 30px, the computer is made to perform the left mouse button click using the pyautogui Python package, as shown.

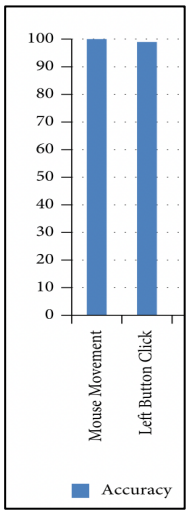




Experimental Results and Evaluation

Hand tip gesture*	Mouse function performed	Success	Failure	Accuracy (%)
Tip ID 1 or both tip IDs 1 and 2 are up	Mouse movement	100	0	100
Tip IDs 0 and 1 are up and the distance between the fingers is <30	**Left button click**	99	1	99





Source Code of The Project

```
import mediapipe as mp
import cv2
import numpy as np
from mediapipe.framework.formats import landmark_pb2
import time
from math import sqrt
import pyautogui

mp_drawing=mp.solutions.drawing_utils
mp_hands=mp.solutions.hands
click=0

video=cv2.VideoCapture(0)
with mp_hands.Hands(min_detection_confidence=0.8,min_tracking_confidence=0.5) as hands:
    while video.isOpened():
        _,frame=video.read()
        image=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
        image=cv2.flip(image,1)
        imageHeight,imageWidth,_=image.shape
        results=hands.process(image)

        image=cv2.cvtColor(image,cv2.COLOR_RGB2BGR)

        if results.multi_hand_landmarks:
            for num,hand in enumerate(results.multi_hand_landmarks):

mp_drawing.draw_landmarks(image,hand,mp_hands.HAND_CONNECTIONS,mp_drawing.DrawingSpec(color=(255,44
,250),thickness=2, circle_radius=2),)

        if results.multi_hand_landmarks !=None:
            for handLandmarks in results.multi_hand_landmarks:
                for point in mp_hands.HandLandmark:

                    normalizedLandmark=handLandmarks.landmark[point]

pixelCoordinatesLandmark=mp_drawing._normalized_to_pixel_coordinates(normalizedLandmark.x,normalizedLand
mark.y,imageWidth, imageHeight)

                    point=str(point)

                    if point=="HandLandmark.INDEX_FINGER_TIP":
                        try:
                            indexfingertip_x=pixelCoordinatesLandmark[0]
                            indexfingertip_y=pixelCoordinatesLandmark[1]
                            pyautogui.moveTo((indexfingertip_x,indexfingertip_y))

                        except:
                            pass

                    elif point=="HandLandmark.THUMB_TIP":
```

```
                    try:
                        thumbfingertip_x=pixelCoordinatesLandmark[0]
                        thumbfingertip_y=pixelCoordinatesLandmark[1]

                    except:
                        pass

                try:
                    Distance_x=sqrt((indexfingertip_x-thumbfingertip_x)**2+(indexfingertip_x-thumbfingertip_x)**2)
                    Distance_y=sqrt((indexfingertip_y - thumbfingertip_y)**2 + (indexfingertip_y -
thumbfingertip_y)**2)

                    if Distance_x < 5 or Distance_x < -5 :
                        click+=1
                        if click%5==0:
                            print("Single Click")
                            pyautogui.click()

                except:
                    pass

        cv2.imshow("Hand Tracking", image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

video.release()
cv2.destroyAllWindows()
```

Project Report

This project is done by me (Gaurav Haritas) and my co-ordinator Vedika Yadav.

The project is done successfully and start tracking the hand and the finger tips then after that our program starts controlling the cursor of the laptop or pc

Conclusion

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions.

From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse.