FUNDAMENTALS OF AI

INCREMENT – 1

HAND DETECTION TRACKING

ABSTRACT:

This paper has been proposed in the view of eliminating the concern of getting contacted with contagious diseases with use of public access computers and to improve the experience of interacting with the computers. Over the years, human-computer interaction (HCI) has seen a tremendous transformation. A HCI interface for Mouse Tracing by Hand Detection is presented in this work. The goal is to use hand motions caught by the webcam to gain control of the mouse and functionality. The users can move the mouse cursor's direction and perform all mouse operations with their hand positioned in front of the webcam.

Introduction:

Direct use of hands is a fundamental way for humans to communicate with each other and, more recently, with gadgets in intelligent surroundings, hence vision-based hand gesture detection is an active topic of research in human-computer interaction (HCI). The trend in human-computer interaction is heading toward real-time hand gesture recognition and tracking for usage in video games, remote-free television control, and other similar applications. Given the increased availability of mobile devices with integrated cameras, such as smart phones and notebook computers, a hand gesture detection system can be a valuable tool for engaging with these camera-enabled devices more naturally than traditional interfaces. This work presents the implementation and analysis of a vision-based static hand position estimate system.

1. **Problem Specification:**

   If you wish to buy a parking ticket or a movie ticket, what if a person infected with a fatal infectious disease previously used it?

If you're delivering a presentation using a projector and want to move the slide but don't have a laser pointer, do you have to walk all the way to the podium and then change the slide?

There are a variety of instances in which this gesture control is required to make the job safe and smart.

2. **Problem Analysis:**

Understanding the above-mentioned real-world problems and the needs of a user to deal with, we propose a solution where in using a web camera and fingertip tracking, users may remotely control their computer mouse with their bare hands. A python library called Mediapipe is being used as an instrumental tool in extracting the hand region, which is then filtered using morphological opening processes and blob labelling. The convex hull and convexity defect are then utilized to count the fingers and determine the fingertip locations. The fingertip coordinates are then transferred to the screen coordinates and smoothed using the Moving Average. Finally, the events associated with the identified fingers are relayed to the computer system to control the mouse. The experimental findings reveal that the suggested approach can count the finger with 98.3 percent accuracy and perform effectively in real time.

DESIGN AND MILESTONES

2.1. Proposed Method
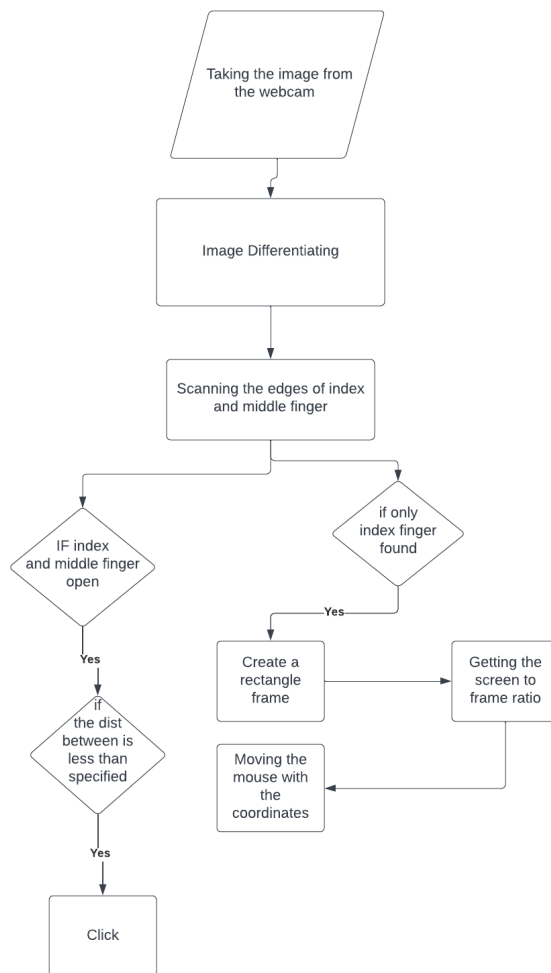
The suggested solution makes use of a few Python packages, including open cv, media pipe, autopy, and pyautogi.

To begin, in order to read the picture from the camera, we utilise the OpenCV library.

The picture is then converted from BGR to RGB using the opencv function cvtColor() as a Media pipe library   Hands is a function that takes an RGB

p i c t u r e   a n d   r e t u r n s   t h e   h a n d   l a n d m a r k s .

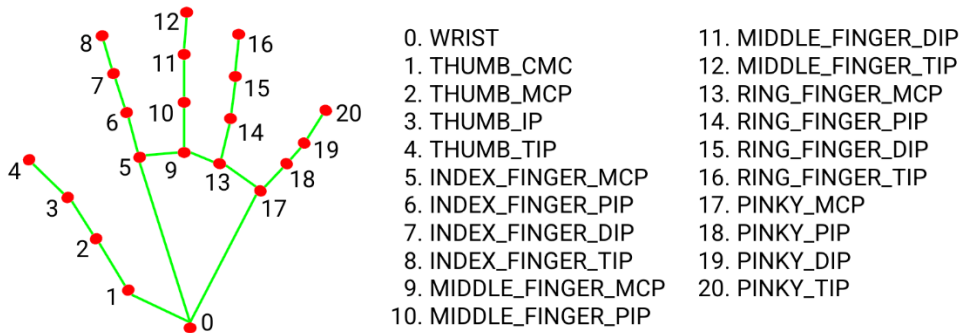| 0. WRIST	11. MIDDLE_FINGER_DIP
1. THUMB_CMC	12. MIDDLE_FINGER_TIP
2. THUMB_MCP	13. RING_FINGER_MCP
3. THUMB_IP	14. RING_FINGER_PIP
4. THUMB_TIP	15. RING_FINGER_DIP
5. INDEX_FINGER_MCP	16. RING_FINGER_TIP
6. INDEX_FINGER_PIP	17. PINKY_MCP
7. INDEX_FINGER_DIP	18. PINKY_PIP
8. INDEX_FINGER_TIP	19. PINKY_DIP
9. MIDDLE_FINGER_MCP	20. PINKY_TIP
10. MIDDLE_FINGER_PIP	

Illustration of Hand Landmarks

Using the landmark, we get the fingertips, which are easy to recognise.

We also draw a rectangle to keep our finger from moving outside the scope of the screen and to make it easier to reach the screen's edges.

We acquire access to mouse the pointer from the specified coordinates by utilising the library called autopy or pyautogui, and if the index and middle fingers get near, it executes the click function.

Validation Methods:

1.Trial and Error:

As here we have to check if the functionality is working or not, we go with trial and error

Future Work:

Addition of more gestures on the thorough analysis of the utility

Addition of Voice Over commands

References:

Computer Vision and Machine Learning based Hand Gesture Recognition Paperback – Import,

12 May 2015 by Paulo Trigueiros (Author), Fernando Ribeiro (Author), Luís Paulo Reis (Author)