FUNDAMENTALS OF Artificial Intelligence
(CSCE 5210)

HAND DETECTION TRACKING
INCREMENT - 2

Team members

Bharath Siva Kumar Yelamanchili

Harsha Kumar Pulla

Kaushic Kolla

Nagaraju Obbineni

ABSTRACT:

This project 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. 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. To the extension of the above-mentioned aim and as an enhancement in the usage of the model, this increment focus on the addition of Voiceover to the mouse movements on screen.

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:

1.1 Data Set:

As this is a system interactive live project, there is no specific dataset that can be used to implement this project. The system will calculate the actions as per the gestures that are created by out hand.

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.

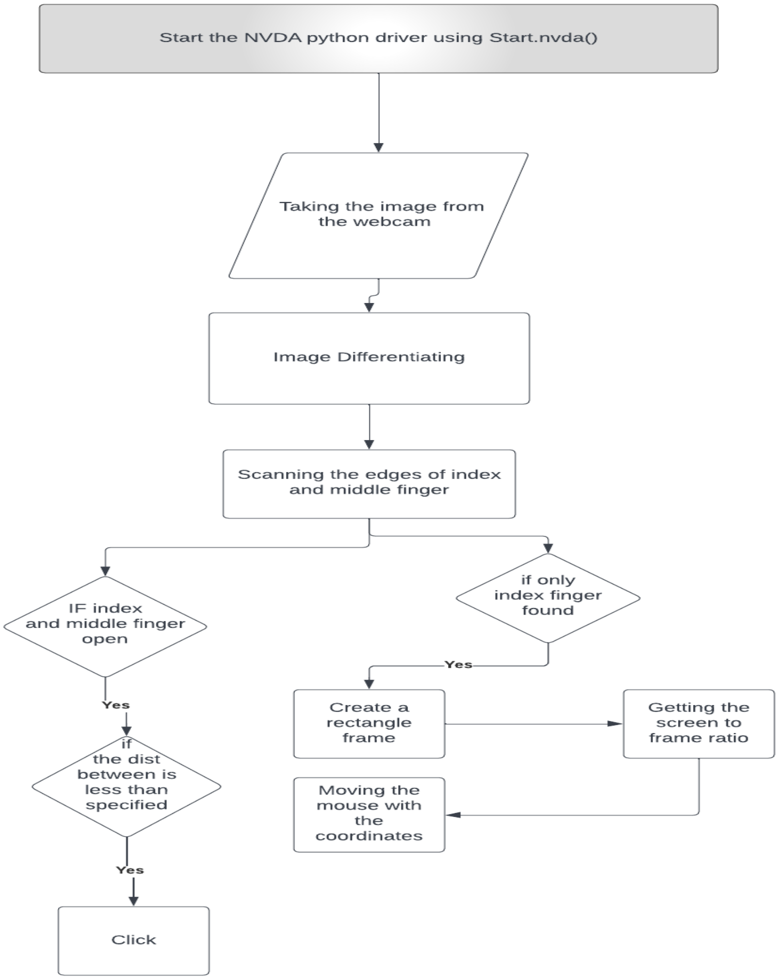
1.2 Problem Analysis:

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.

2. DESIGN AND MILESTONES

2.1 Proposed Method



Start the NVDA python driver using Start.nvda()
Taking the image from the webcam
Image Differentiating
Scanning the edges of index and middle finger
IF index and middle finger open
Yes
if the dist between is less than specified
Yes
Click
if only index finger found
Yes
Create a rectangle frame
Getting the screen to frame ratio
Moving the mouse with the coordinates

Fig: Shaded region refers to Increment 2

2.2 Data Processing

As this is system interactive project, we do not have a specific dataset that can be used and demonstrate the project however, the below are the steps we followed to implement the project and how the system takes the hand gestures as an input:

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

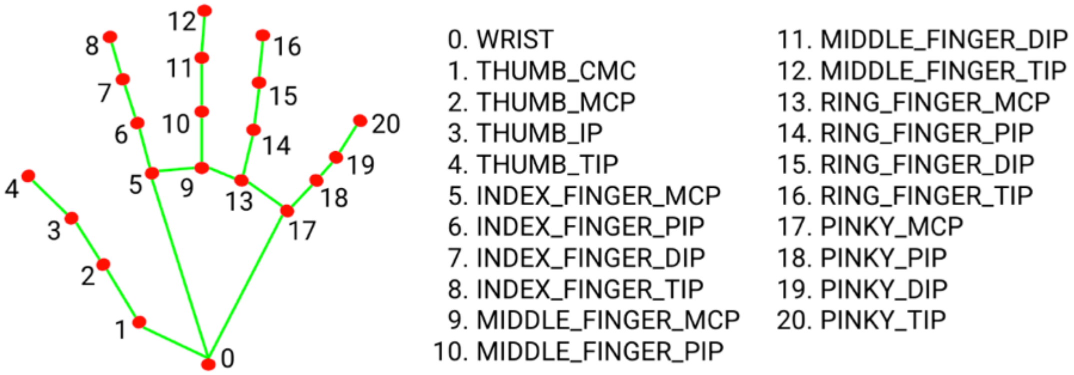
b. Initially, to read the picture from the camera, we utilize the OpenCV library.

c. 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 picture and returns the hand landmarks.

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

e. 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.

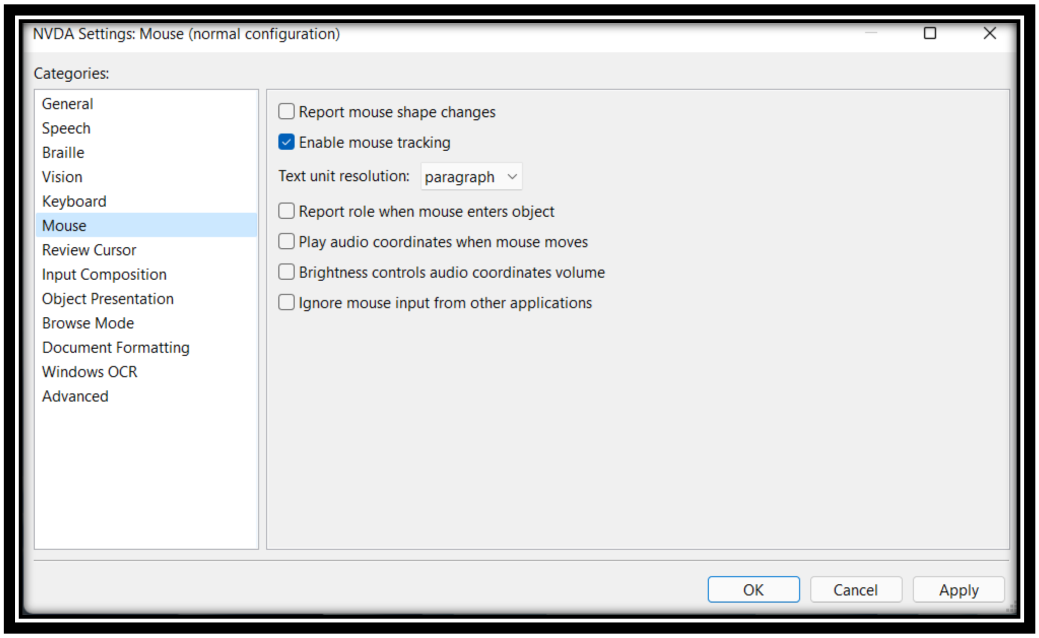
f. We acquire access to mouse pointer from the specified coordinates by utilizing the library called autopy or pyautogui, and if the index and middle fingers get near, it executes the click function.



2.3 Experimental Settings

In this increment, the solution is to integrate Voiceover commands to aid in clear navigation across the screen. It also aids people with low vision or blindness to access the system.

As the shaded region highlights to start NVDA driver, which is a product of NV Access organization. This driver is created to lower the economic and social barriers associated with accessing Information Technology for people who are Blind or Vision Impaired.



□ Report mouse shape changes
☑ Enable mouse tracking
Text unit resolution: paragraph
□ Report role when mouse enters object
□ Play audio coordinates when mouse moves
□ Brightness controls audio coordinates volume
□ Ignore mouse input from other applications

After installing the driver, it asks for the preferences to set, as our application involves mouse movements to be captured, make sure the mouse tracking is checked as shown in the above figure.

When enabled, NVDA will announce the text currently under the mouse pointer, as you move it around the screen. This allows you to find things on the screen, by physically moving the mouse, rather than trying to find them through object navigation.

2.4 Validation Methods:

Trial and Error:

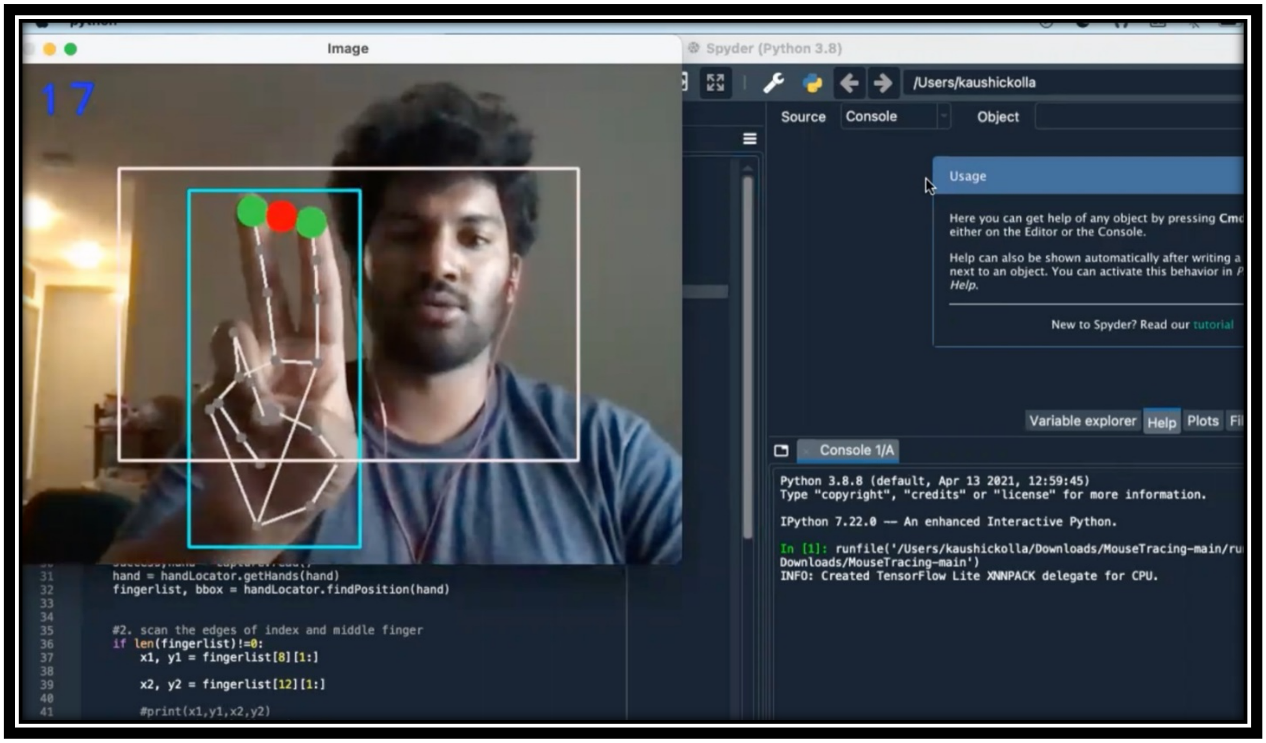
We used trial and error methodology to validate the functionality of mouse detection system by hovering over various icons and access various applications and performing certain activities like navigating through different icons opening different files, and browsers that we usually perform using the traditional mouse.

In addition to the above testing, we also performed rigorous testing to ensure that system was able to read the regular file types and the text saved in the documents as we hover over it.

2.5 Results and Analysis

We were able to achieve the main objective of the project. The mouse control using the hand detection gestures has worked with 100% of accuracy in windows operating system and with 90 to 95% accuracy in Mac operating system. It seems simple to use for the end user and it is done through scalable code as well.

The below is the sample screenshot from Demo we performed to explain how the hand detection mechanism works to hover over the mouse.



Here you can get help of any object by pressing Cmc
either on the Editor or the Console.

Help can also be shown automatically after writing a
next to an object. You can activate this behavior in *P*
Help.

New to Spyder? Read our tutorial

```
hand = handLocator.getHands(hand)
fingerlist, bbox = handLocator.findPosition(hand)


#2. scan the edges of index and middle finger
if len(fingerlist)!=0:
    x1, y1 = fingerlist[8][1:]

    x2, y2 = fingerlist[12][1:]

    #print(x1,y1,x2,y2)
```

```
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/kaushickolla/Downloads/MouseTracing-main/ru
Downloads/MouseTracing-main')
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

Future Work:

We would like to add few more additional gestures and perform a thorough analysis of utility addition of voice over commands. In addition to this we would like to make the system interactive not limiting the mouse controlling system to just open, close, or select certain files, we want to use the hand-gestures movements to use the mouse pointer to scribble in a word or text document and want to implement a machine learning algorithm to convert the irregular text into a regular format.

3.1 Implementation status report

3.1.1 Work Completed:

We imported an opensource python file that facilitates us to implement a part of the project proposed. We used open cv and media pipeline libraries to commit the project. These two libraries will the main part to run the project. They capture gestures of the hand to move the mouse cursor on the screen.

3.1.1.2 Responsibilities:

We made our tasks clear and distributed the roles among ourselves according to skillset and knowledge of the team members.

Role	Name
Integration of camera code	Kaushic Kolla
Development of opencv code	Nagaraju Obbineni
Development media pipe	Bharath Siva Kumar Yelamanchili
Voice Integration and testing	Harsha Kumar
Documentation	All

3.1.1.3 Contributions

Everyone in the team has made an equal contribution in implementation of this project. Every module implemented by the team member is interdependent with another module, a failure in one module affects the other modules in a row.

Name	
Kaushic Kolla	25%
Nagaraju Obbineni	25%
Bharath Siva Kumar Yelamanchili	25%
Harsha Kumar Pulla	25%

3.1.1.4 - Issues and concerns

We have developed the OpenCV code without any issues however we faced certain challenges while we tried to integrate voice when we hover over. To overcome this challenge, we imported an open source that is used by blind and deaf people and integrated with our code.

The functionality is working with an accuracy of 100 percent in the Windows operating system, but the accuracy was between 85 and 90 in Mac operating system while the mouse is hovered over the icon.

Project GitHub Link:

https://github.com/kaushic-kolla/mouse_detection/blob/main/run.py

References:

[1] Murthy, G.R.S.; Jadon, R.S. A review of vision-based hand gestures recognition. Int. J. Inf. Technol. Knowl. Manag. 2009, 2, 405–410.

[2] Suriya, R.; Vijayachamundeeswari, V. A survey on hand gesture recognition for simple mouse control. In Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 27–28 February 2014; pp. 1–5.

[3] Hasan, M.M.; Mishra, P.K. Hand gesture modeling and recognition using geometric features: A review. Can. J. Image Process. Comput. Vis. 2012, 3, 12–26

[4] Perimal, M.; Basah, S.N.; Safar, M.J.A.; Yazid, H. Hand-Gesture Recognition-Algorithm based on Finger Counting. J. Telecommun. Electron. Comput. Eng. 2018, 10, 19–24.

[5] Fang, Y.; Wang, K.; Cheng, J.; Lu, H. A real-time hand gesture recognition method. In Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; pp. 995–998.

[6] Van den Bergh, M.; Koller-Meier, E.; Bosché, F.; Van Gool, L. Haarlet-based hand gesture recognition for 3D interaction. In Proceedings of the 2009 Workshop on Applications of Computer Vision (WACV), Snowbird, UT, USA, 7–8 December 2009; pp. 1–8.

[7] Pun, C.-M.; Zhu, H.-M.; Feng, W. Real-time hand gesture recognition using motion tracking. Int. J. Comput. Intell. Syst. 2011, 4, 277–286

[8] Chen, Y.; Luo, B.; Chen, Y.-L.; Liang, G.; Wu, X. A real-time dynamic hand gesture recognition system using kinect sensor. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 2026–2030.

[9] Bamwenda, J.; Özerdem, M.S. Recognition of Static Hand Gesture with Using ANN and SVM. Dicle Univ. J. Eng. 2019, 10, 561–568.

[10] Bao, P.; Maqueda, A.I.; del-Blanco, C.R.; García, N. Tiny hand gesture recognition without localization via a deep convolutional network. IEEE Trans. Consum. Electron. 2017, 63, 251–257.

[11] John, V.; Boyali, A.; Mita, S.; Imanishi, M.; Sanma, N. Deep learning-based fast hand gesture recognition using representative frames. In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016

[12] Lee, D.-H.; Hong, K.-S. Game interface using hand gesture recognition. In Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology, Seoul, Korea, 30 November–2 December 2010; pp. 1092–1097.

[13] Wan, C.; Probst, T.; Van Gool, L.; Yao, A. Self-supervised 3d hand pose estimation through training by fitting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10853–10862

[14] Dinh, D.-L.; Kim, J.T.; Kim, T.-S. Hand gesture recognition and interface via a depth imaging sensor for smart home appliances. Energy Procedia 2014, 62, 576–582.
[15] Guan, Y., Zheng, .M. (2008). "Real-time 3D pointing gesture recognition for natural HCI. IEEEProceedings of the 7th World Congress on Intelligent Control and Automation WCICA 2008, doi:10.1109/WCICA.2008.4593304

[16] Cheng-Chang L. and Chung-Lin H., (1999)."The Model-Based Dynamic Hand Posture IdentificationUsing Genetic Algorithm", Springer, Machine Vision and Applications Vol. 11

[17] Minghai Y., Xinyu Q., Qinlong G., Taotao R., Zhongwang L., (2010). "Online PCA with AdaptiveSubspace Method for Real-Time Hand Gesture Learning and Recognition", journal World Scientificand Engineering Academy and SocietWSEAN, Vol. 9(6)

[18] Verma, R., Dev A. (2009)."Vision based hand gesture recognition using finite state machines andfuzzy logic". IEEE International Conference on Ultra-Modern Telecommunications & Workshops(ICUMT '09), pp. 1-6. doi: 10.1109/ICUMT.2009.5345425.

[19] Malima, A., Özgür, E., Çetin, M. (2006). "A Fast Algorithm for Vision-Based Hand GestureRecognition For Robot Control", IEEE 14th conference on Signal Processing and CommunicationsApplications, pp. 1-4. doi: 10.1109/SIU.2006.1659822

[20] Mahmoud E., Ayoub A., J¨org A., and Bernd M., (2008). "Hidden Markov Model -Based Isolatedand Meaningful Hand Gesture Recognition", World Academy of Science, Engineering andTechnology 41

Open-Source files:

[14] https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-020-4916-6

[15] https://link.springer.com/article/10.3758/s13428-019-01258-6

[16] http://pascalkieslich.github.io/mousetrap/

[17] https://github.com/pascalkieslich/mousetrap-resources.