Human Computer Interface (HCI) controlled AI enabled System for Optimization

[1]Hitesh Kumar Sharma *!, [2]Tanupriya Choudhury *!, [3]Rakesh Soni, [4]Shilpi Sharma
[1,2]School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun,248007,Uttarakhand, India.
[3]Genpact India Private Limited, Gurgaon, 122002,Haryana, India.
[4]Dept. of CSE, Amity University Uttar Pradesh, 201301, India.
hkshitesh@gmail.com*, tanupriya@ddn.upes.ac.in*, rakesh.soni1998@gmail.com, ssharma22@amity.edu
*Corresponding Author(s), ! marked authors are contributed equally and are the first authors.
[1]-IEEE Member(ID-93317822), [2]-IEEE Senior Member (ID-94931644)

Abstract—In recent years, people got more interested in the research of the evolution of human-computer interaction (HCI). Right from the transition of a text-based command line interface to a graphical user interface (GUI) that uses a mouse, many changes have been observed. The ability to manipulate the mouse with hand movements is one of the areas of interest here. At present, due to the cluttered backgrounds and poor lighting, this is a complicated issue. The application demands extremely specific detection and recognition. In this article, a computer vision-based method has been shown for controlling a computer mouse with hand gestures in front of a linked camera. The user may move the mouse and use the left as well as right click functions using this systematic program.

Keywords—Human Computer Interface (HCI), Deep Learning, Hand Gesture.

I. Introduction

In today's world, computers have become a day-to-day necessity for humans even for the smallest tasks. To make our work easier, it needs to interact with computer ample number of times a day, to make our work easier. This led to the emergence of a hot topic for research, the Human Computer Interaction. A computer's mouse is extremely helpful tool that makes it much easier for humans to interact with the machine. But the problem occurs when a person with a physical disability uses the mouse and as a result, while using a mouse can be challenging in situations, it may be impossible for others depending on the severity of the disability. To solve this problem, a system that uses a gesture recognition technology is developed to provide mouse control (GRS). The inconvenience of using a mouse while travelling is another issue that will be addressed while developing GRS, thus a better mechanism must be put in place to manage it. Humans rely heavily on their senses for their communication. Similar to this, a computer's Graphical User Interface (GUI) heavily relies on the mouse.

By blending both of these approaches in the field of digital signal processing, a more effective interactive system for human-computer interaction is formed[8][9][10]. The quantity of information included in an image signal can be used to control a variety of computer operations. Hand gestures are therefore vital in this subject. In this research, a method for controlling the mouse with hand gestures is offered in front of a system camera utilizing computer vision. Python libraries are employed, which are excellent tools for working with images and modifying them as well as automating computer programs like OpenCV and patagium. The user needs only the laptop's web camera to use this; no colored markers are required. For hand detection, this method does not require any dots or bands on the fingers or wrists. The user intends to scroll the mouse in the direction of the fingers' movement when three fingers are raised and spaced apart. If the fingers are pressed together, the user can move his hand without moving the mouse if he/she chooses. The user must lift 4 fingers to click left and 5 fingers to click right, respectively. In this article, a thorough study has been done on literature, a detailed description is given in dataset portion, the mathematical formulation of model has been done, along with the methodology and experimentation results has been discussed in this paper. Last but not the least the paper is concluded in an effective manner.

II. Literature Review

In past technologies, hand gesture detectors required to employ some sort of mechanical device to get data about the gesture. One illustration of such gadgets is a data glove device [2], which allows users to input data into a computer system by moving their fingers. Although the data glove was fairly popular in the past, advances in computer hardware development over the past ten years have allowed for the manufacturing of cheaper hardware while also improving computing performance. Data gloves are gradually being replaced with vision-based hand gesture techniques, which are more natural and user-friendly because they don't require any wearable technology. One of the most crucial elements of human-computer interaction is this. The data glove has a bulky appearance and restricts how well the hand can move. One of the six tangible forms that computers must take in order to be perceptibly transmitted to humans is vision [4]. Therefore, in order to recognise hand gestures, a vision-based technique is preferable and friendlier than wearable technology. To control mouse movement via video devices, numerous techniques have been developed. The authors of [5] attempted to utilise a camera with a colour detection approach to control the mouse pointer. To give the system information, the user must, however, wear coloured tapes. In [4], the color-based segmentation technique is used for gesture control.

The use of red gloves in this case facilitates segmentation but has the disadvantage of increasing costs and possibly causing pain to users. In [5] , colour pointers were utilised. In this case, "flipping of image" was required. The use of coloured pointers could be uncomfortable and more expensive. Another piece of work is by Alisha Pradhan and B.B.V.L. Deepak [7], who created an interface for using a computer without a mouse. However, they are making the hand signals while wearing colored gloves. Using any gloves is avoided in our operations. Instead, the mouse is operated with just our bare hands and less complicated techniques. A novel method of hand gesture recognition through the use of input-output technology Hidden Markov Models were proposed by Marcel [5] and his colleagues. It required tracking the skin-color blobs of the human body in order to recognize two classes of movements. Using a hidden Markov model approach, Chen and colleagues trained the hand gesture to recognize the hand positions. The hidden Markov model needs more work to train on hand motions than Cascade classifiers do. Numerous researchers' recent research has shown that the hand gesture system is not only theoretically and technically solid, but also simple to integrate into a variety of application systems and environments.

III. DATASET DETAILED DESCRIPTION

Controlling a physical mouse could be sometimes tiring. It can be inconvenient to use it while travelling. In such circumstances, it will be undesirable to use hardware based hand gesture recognition where the user must wear a device. Therefore, this problem is taken up and a method is presented which does not require any external device to control the mouse. In these times of COVID-19, when it can be contagious to touch any surface or object, such a method would be a boon as it would not require to touch anything. The computer would be able to be controlled by making hand gestures in the air.

The Hand Gesture Recognition Database is used from]https://www.kaggle.com/code/benenharrington/hand-gesture-recognition-database-with-cnn/data] [6] for this project. On Kaggle, this dataset is accessible. It includes 12,000 photos of various hand gestures made by various ethnic groups. This dataset includes 6 different people's hand movements. In this dataset, several peoples are taken into account as subjects.

Therefore, 3 male and 3 female individuals were chosen to record their various hand movements. The Leap Motion hand tracking device was used to take the pictures. (Table 1)

Table 1. –Dataset detailed description

No. of Images	12000
No. of Peoples Involved	6 (03 Female and 03 Male)
Types of Hand Gestured Recorded	10 (table 3)
Total No. of Classes	05
Dataset Dim.	640*240
Sampled Used	1800
Training Images	1500
Testing Images	300

Only about 10% of the 12000 photos is sampled in a huge dataset (2010) in order to train our CNN model. Due to limited computing resources, only sampled 10% of the photos is taken. One person sample is selected from a total of six different person samples. All ten different varieties of a single person's hand gestures are included in the example dataset. Each type includes 200 photos shot from various angles. The various hand gestures are mapped with label numbers in table 3. From 01 to 05, Five different hand motions are recorded and identified . According to the information in the accompanying table, the suggested model has been trained to categorize a set of photos into classes ranging from 01 to 05. (Table 2)

Table 2: Label Used for Hand Gesture

Type of Gesture (Hand)	Label Defined
Thumb (Down)	01
Horizontal Palm	02
Thumb (Up)	03
Index Finger	04
Ok	05

Due of resource constraints, just 10% of the images—or the 2010 images—were sampled in order to train our model. Figure 1 displays six randomly chosen photos from a pool of images. These pictures display a single person's six varied hand movements.

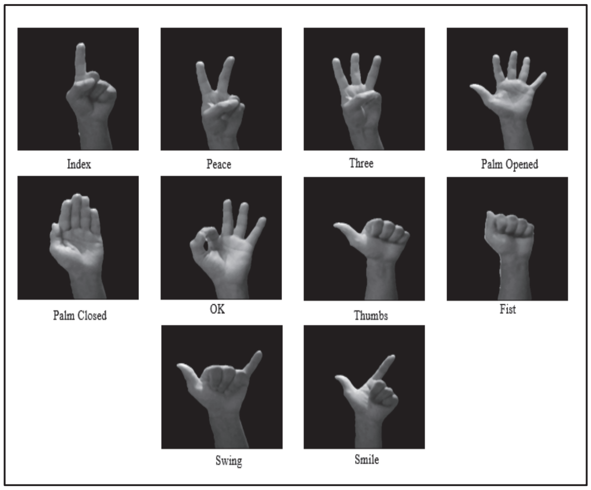

Figure 1: Randomly selected images from training sample

IV. MATHEMATICAL FORMULATION OF MODEL

Any neural network must go through the following two crucial steps in order to be trained:

- Using forward propagation, data is received, processed, and output is produced.
- Calculate the error and update the network's parameters for backward propagation.

An asterisk * symbol is frequently used in mathematics to denote convolution. The expression would be: The symbol used for image is X and for filter is f. The the formula for

output image will be denoted as. (eq.1)

$$Z = X * f \quad \text{------}(1)$$

- Image Dimension= (n, n)
- Filter Dimension = (f,f)
- Output image dimension can be find as ((n-f+1) , (n-f+1))

The data is transferred to the fully linked layer after it has been converted into a 1D array. Linear and Non-liner operation is performed using fully connected layers. Fully connected layer is used to convert the 2D matrix into 1D vector. Liner transformation can be computed as following equation. (eq. 2)

$$Z = W^T * X + b \text{ --------}(2)$$

Given that the matrix's dimension is (m, n), this layer will have exactly m features or inputs. Since there are 4 features in the convolution layer, the value of m in this situation would be 4. The number of neurons in present layers is used to find the value of n. For instancefor two neurons following calculation can be done: (4, 2):

Now that the weight and bias matrices have been defined, let us add them to the equation for linear transformation.:

$$X = \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} \qquad W = \begin{bmatrix} w11 & w12 \\ w21 & w22 \\ w31 & w32 \\ w41 & w42 \end{bmatrix}$$

Input Data Randomly initialized weight matrix

$$b = \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

Randomly initialized bias matrix

Matrices for X, W and b is given above.

With eq. 3 the matrices for parameters, weights and bias has been given.

The Sigmoid activation function will be used to solve a binary classification issue. Let's The mathematical formula is given below. (eq. 3)

$$Z = W^T . X + b \quad \text{--------} (3)$$

$$Z = \begin{bmatrix} w11 & w12 & w13 & w14 \\ w21 & w22 & w23 & w24 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} + \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

$$Z_{2x2} = \begin{bmatrix} w11x1 + w12x2 + w13x3 + w14x4 \\ w21x1 + w22x2 + w23x3 + w24x4 \end{bmatrix}$$

In eq. 4 output produce by a neuron is given.

$$f(x) = 1/(1 + e(pow)(-x) \text{ -------}(4)$$

A Sigmoid function has a range of 0 to 1. This implies that the outcome would always fall within the range for whatever input value (0, 1). The sigmoid function is applied, which is frequently used for binary classification issues, for both convolutional and fully-connected layers.

V. METHODOLOGY AND ITS IMPLEMENTATION

In this research work the hand gestured dataset is loaded in jupyter notebook for training the proposed model. First it was converted into grayscale then size will be rescaled as per model requirement. Python functions have been developed to convert, rescale and process images for training the model. The step by step process for the proposed methodology is given below.

- Capturing real time video using a web camera.
- Processing each image frame.
- Flipping of each image frame.

- Conversion of frame to a grayscale image.
- Finding contours and drawing the area.
- Counting the number of convexity defects.
- Moving the mouse cursor using pyautogui moveRel function when number of convexity defects is equal to 2.
- Implementing left click and right click by using pyautogui.click() and pyautogui.rightClick() respectively.

To capture the video from the web camera of the laptop, use of PyAutoGUI library in Python is done. It is a cross-platform GUI automation module.

Capturing Video

```
cap = cv2.VideoCapture(0)
while cap.isOpened():
# other code
```

To pre-process each frame read by the system, a widely used library for image processing, OpenCV is used.

Preprocessing frames

```
img = cv2.flip(img, 1)
copy_img = img
grayVAl = cv2.cvtColor(copy_img,
cv2.COLOR_BGR2GRAY)
value = (35, 35)
blurValue = cv2.GaussianBlur(grayVAl, value, 0)
thval, thresh1Val = cv2.threshold(blurValue, 127, 255,
cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
```

Plot the contours and initialize the variable defects Number with 0 and increment the value for every time when the angle is less than 90 degrees. Then compare the defects Number. If it is equal to 2, store the frame points and make the mouse cursor move, if it is equal to 3, perform left click and if 4 perform right click .

Implementing clicks

```
if defectsNumber == 2 and angle <= 90:
        defectsDone = {"x": spoint[0], "y": spoint[1]}
For right and right click:
elif defectsNumber == 3 and CLICK is None:
    pyautogui.click()
    CLICK_MESSAGE = "LEFT CLICK"
elif defectsNumber == 4 and CLICK is None:
    pyautogui.rightClick()
    CLICK_MESSAGE = "RIGHT CLICK"
```

.

VI. EXPERIMENTAL RESULTS

Table 3 lists the parameters for the CNN model. The following table lists the called epochs and the performance metrics for predictions. A good classification model produces accurate results for training time.

Table 3 –Parameters used in Model Training

Parameters for training	Value
No. of Epochs	10
Score for PPV	0.92
Score for Sensitivity	0.93
F1 Score	0.94
Training Accuracy (%)	96.22

The transpose matrix for the proposed model is given figure 4.

	Predicted Thumb Down	Predicted Palm (H)	Predicted Thumbs up	Predicted Index	Predicted OK
Actual Thumb Down	56	0	0	0	0
Actual Palm (H)	0	47	0	0	0
Actual Thumbs up	0	0	55	0	0
Actual Index	0	0	0	52	0
Actual OK	0	0	0	0	58

Figure 4: Proposed Model Transpose Matrix

12 test images is fed into the new model (figure 3) as test inputs and compared the output with labelled data to test the prediction output. The outcomes are displayed in figure 4,

figure 5, figure 6 and figure 7. The trained model's output is plotted using the Python tool matplotlib.

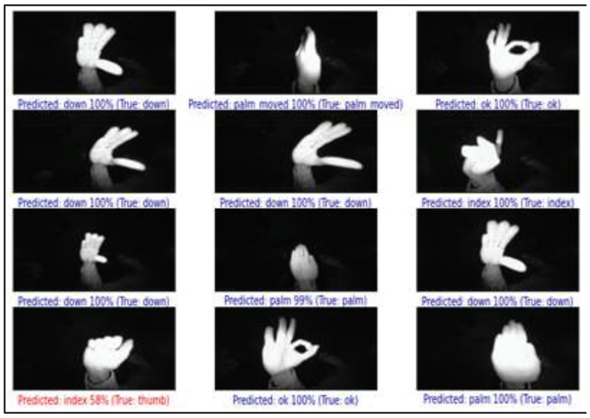

Figure 5: Hand Gesture prediction using Proposed CNN Model

In figure 5, the prediction of hand gesture images if shown. Out of 12 only one image is predicted false.

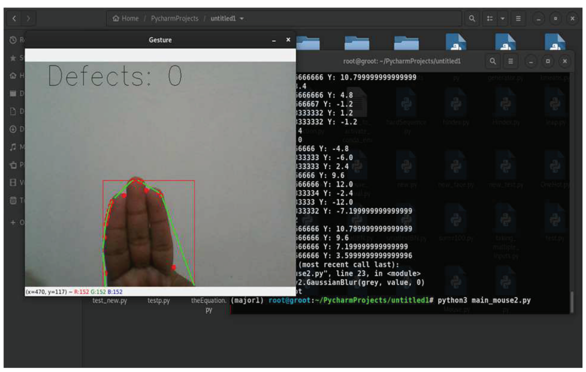

Figure 6: No movement of mouse

In figure 6, it has been shown that no action will be taken on providing the shown hand gesture.

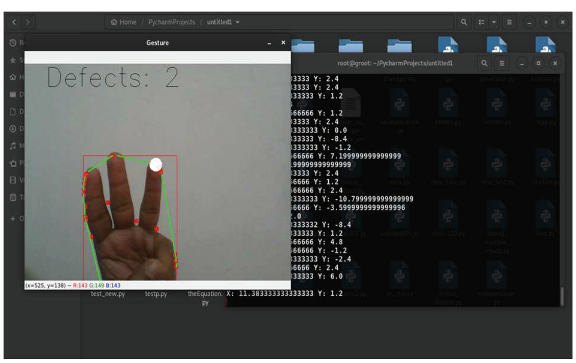

Figure 7: Scroll the mouse using 3 fingers like this

In figure 7, the mouse scrolling feature will be effective. If the user will use three finger than this gesture is mapped with scrolling feature of mouse.

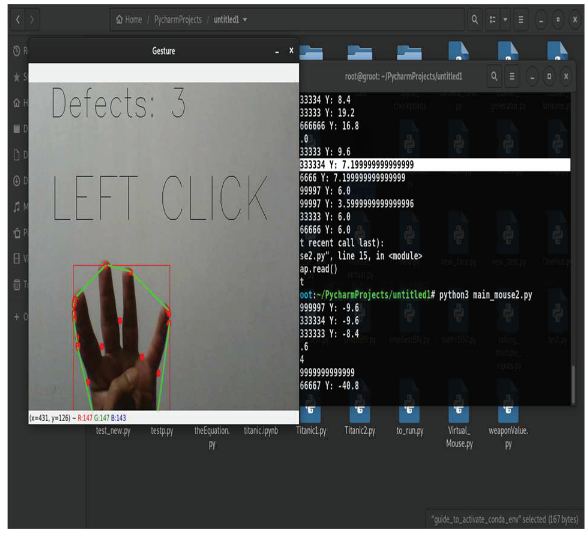

Figure 8: Make a left click by showing 4 fingers of the same hand

In figure 8 and 9, the mouse left click and right click feature will be effective respectively. If the user will use four fingers than this gesture is mapped with left click feature of mouse.

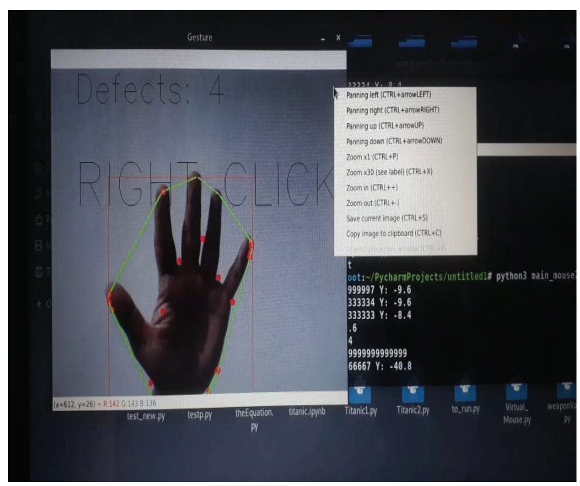

Figure 9: Make a right click by showing 5 fingers of the same hand

If the user will use five fingers than this gesture is mapped with right click feature of mouse.

VII. Conclusion

Thus, it can be said that our suggested technique works effectively and efficiently with just bare hands, the built-in webcam, and no coloured markers or caps. The probability of mouse-less computing is known to everyone. A hand gesture recognition system has many benefits over the conventional touchless interfaces like speech and touch-controlled computers. For instance, like those involving distance and noise, it is also superior to other devices. In the future, it can be used so as to control more complex applications, like multiplayer video games. In addition, to include more functionality like window resizing and enlargement, multi-window interaction, and so on as it is wanted. Increasing the system's sensitivity to backdrop variables like illumination is another update that can be done. Then, it will be able to function properly.

References

[1] Linda Wang, Zhong Qiu Lin, Alexander Wong "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images" Article number: 19549 (2020)

[2] Mais Yasen, Shaidah Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications ", PeerJ Computer Science, Sep 2019.

[3] Nico Zengeler et al., "Hand Gesture Recognition in Automotive Human–Machine Interaction Using Depth Cameras", Sensors 2019, 19(1), 59; https://doi.org/10.3390/s19010059.

[4] T. Mantecón, C.R. del Blanco, F. Jaureguizar, N. García, "Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller", Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi: 10.1007/978-3-319-48680-2_5).

[5] S. Marcel. Hand posture recognition in a body-face centered space. In Proceedings of the Conference on Human Factors in Computer Systems (CHI), 1999.

[6] https://www.kaggle.com/code/benenharrington/hand-gesture-recognition-database-with-cnn/data.

[7] A. Pradhan and B. B. V. L. Deepak, "Obtaining hand gesture parameters using image processing," 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Avadi, India, 2015, pp. 168-170, doi: 10.1109/ICSTM.2015.7225408.

[8] H. Garg, T. Choudhury, P. Kumar and S. Sabitha, "Comparison between significance of usability and security in HCI," 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 2017, pp. 1-4, doi: 10.1109/CIACT.2017.7977269.

[9] N. Parashar, R. Soni, Y. Manchanda and T. Choudhury, "3D Modelling of Human Hand with Motion Constraints," 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 2018, pp. 124-128, doi: 10.1109/CTEMS.2018.8769229.

[10] A. Das, S. C. Gupta, T. Choudhury and S. Sachan, "A fully comprehensive & relative study of Human Brain-Computer Interface Algorithms," 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), Paris, France, 2018, pp. 457-460, doi: 10.1109/ICACCE.2018.8441706.