Hand Gesture Recognition based Virtual Mouse using CNN

Aabha Waichal
Student
Shivaji Nagar
Pune

Mauli Gandhi
Student
Keshav Nagar
Pune

Srushti Bhagwat
Student
Kottur Road
Aheri

Amruta Bhanji
Student
Pawdewadi Naka
Nanded

Shalaka Deore
Assistant Professor
MESCOE
Pune

Shubhangi Ingale
Assistant Professor
MESCOE
Pune

ABSTRACT
Advancements in the field of computer have been a crucial point in the history of mankind. It has proved to be beneficial in many countless fields. Input and output devices which are of utmost importance have also undergone numerous changes. Human Computer Interface (HCI) is an important part of computer systems. Mouse which is one of the input devices is constantly evolving giving rise to various alternatives like touchscreens. But still traditional mouse is prevalent. In this paper, we have developed a virtual mouse which can potentially replace traditional mouse. It detects hand in the live feed, recognizes hand gestures and Convolutional Neural Network (CNN) classifies them and accordingly appropriate mouse operation is done. Hand gestures which make up a crucial part of HCI are employed in this system.

Keywords
Virtual mouse, Hand Gesture Recognition, CNN model.

1. INTRODUCTION
Past few years have witnessed significant advancements and improvements in technology. New technologies have been introduced as well as existing ones were evolved into much better and efficient versions. One of the major targeted portions is of input and output devices. Bluetooth mouse, similar to the wired mouse, was also introduced. Touchscreens have been introduced in hopes to replace traditional mouse. There have been approaches to replace traditional mouse like introducing a glove-based detector [1]. But the notion to replace hardware was not achieved in this approach.

After this mouse based on gesture recognition is being explored. This is based in image processing, recognition and classification. Through this, different gestures are captured, identified and according to that mouse operations are done. For this various image processing techniques [2] have been tried and applied including contour and convex hull area [3].

Pre-processing is an important step in image processing to achieve accurate results. For this, various pre-processing techniques have been employed like k-cosine and border-tracing [4], background subtraction [3], computing four motion matrices [5].

Gesture recognition is the last step in this type of virtual mouse. For this, techniques like 3D convolutional neural network [4] have been discussed. Colored finger caps [6] is also one of the methods used for this so that color identification can be used but again it doesn't satisfy the idea of eliminating requirement of hardware.

In this paper we have developed a system which puts forward an interactive way of controlling the movement of mouse by hand gesture. The in-built webcam is used to capture the live feed. Using python libraries hand detection in the image and background subtraction is done. The pre-processed image is then passed to CNN model for recognizing the gesture and accordingly the cursor is controlled. CNN model is trained by the dataset prepared.

2. METHODOLOGY
The whole system can be majorly divided into 3 sections namely- capturing live feed and pre-processing, CNN model, accessing mouse virtually. The whole system is implemented in python utilizing various libraries like OpenCV, TensorFlow, Keras, PyAutoGUI, NumPy. The architecture of the system is shown in the fig (1).

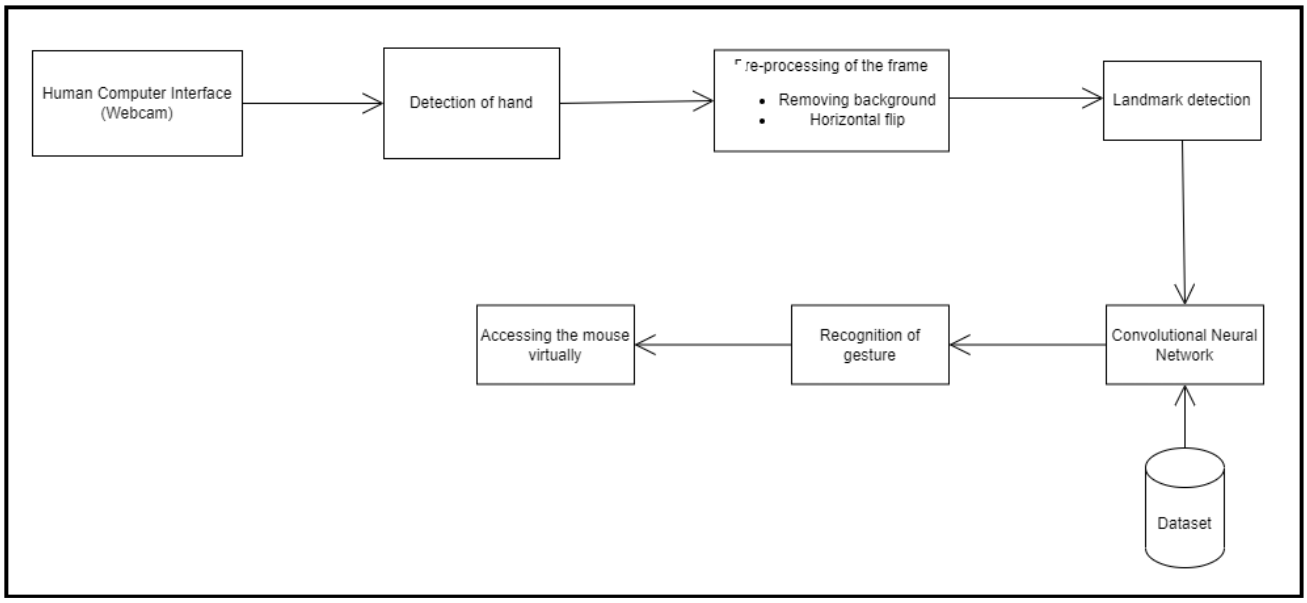

Fig 1: Architecture Diagram

2.1 Capturing live feed and pre-processing

The live feed is captured from the webcam attached to the system. It constantly extracts image from the live video. The extracted image is then flipped horizontally. Once hand is detected, it extracts image from the live feed. The size of the input is 1280 x 720.

Once the image is captured, an empty array of the size of the shape of image is created and it is filled with the numeric equivalent of white i.e., 255. The captured image is then converted from BGR to RGB and tested to see whether hand is detected in the frame or not. If hand is detected, then the subsequent elements of the array are marked for the 21 co-ordinates using Landmark algorithm from MediaPipe framework. Lines are also marked joining these co-ordinates. The live feed which is constantly displayed on the screen for reference is also marked with the same co-ordinates and lines joining the same. Only one hand is detected at a time to prevent confusion and inability to control the mouse.

The co-ordinates are normalized and also stored separately for further use. The array is then resized and converted to image. It is then passed to the CNN for identifying the gesture. The result from CNN is

displayed on the screen as well for the user. The 10th co-ordinate i.e., the co-ordinate at the base of the middle finger is used to move mouse pointer on the screen accordingly. It is the reference point which determines the position of the mouse pointer.

2.2 Dataset

The dataset used in training the CNN model which is used for virtual mouse in this paper is self-made. It consists of total 9000 images. As 6 different operations are included in this mouse, 6 different gestures are required. So, for each class there are 1500 images. The dataset is divided in the ratio of 80-20, as a result, 1200 images for each gesture are used for training and 300 images are used for testing for each gesture. These images are preprocessed similar to the images that will be passed to CNN while using the system. The images have a white background and the 21 co-ordinates on the hand are marked with circles and these are joined by lines. While collecting dataset, care has been taken so that variations of the gestures are included to increase the scope. This means the gestures in some images are titled, have different positions so that the diversity is maintained. As the dataset contains images of right hand, at this point, the virtual mouse works can be controlled accurately using right hand only. The images below are example of each of the gestures.

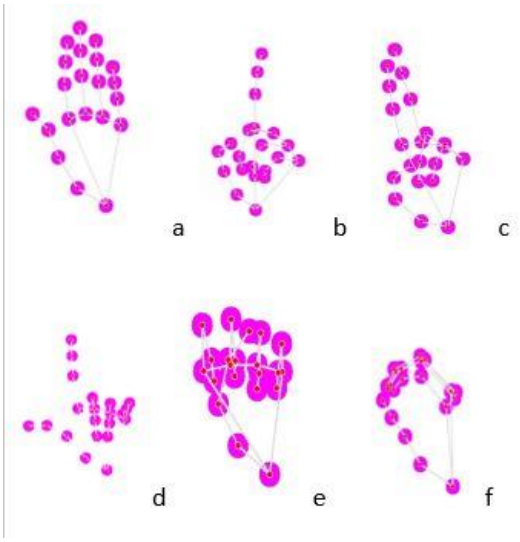

Fig 2: Dataset a. navigationb. left clickc. right clickd. scroll up e. scroll down f. screenshot

2.3 CNN Model

The CNN model which is trained is of sequential type. That means the model is built layer by layer.

Training and testing batches are made and data augmentation is done with the help of ImageDataGenerator. The batch size is kept as 32 and it is shuffled. While pre-processing, the images are converted into arrays. The images are resized to 64 x64. The class mode is set to categorical and not binary as the images will classified into different classes.

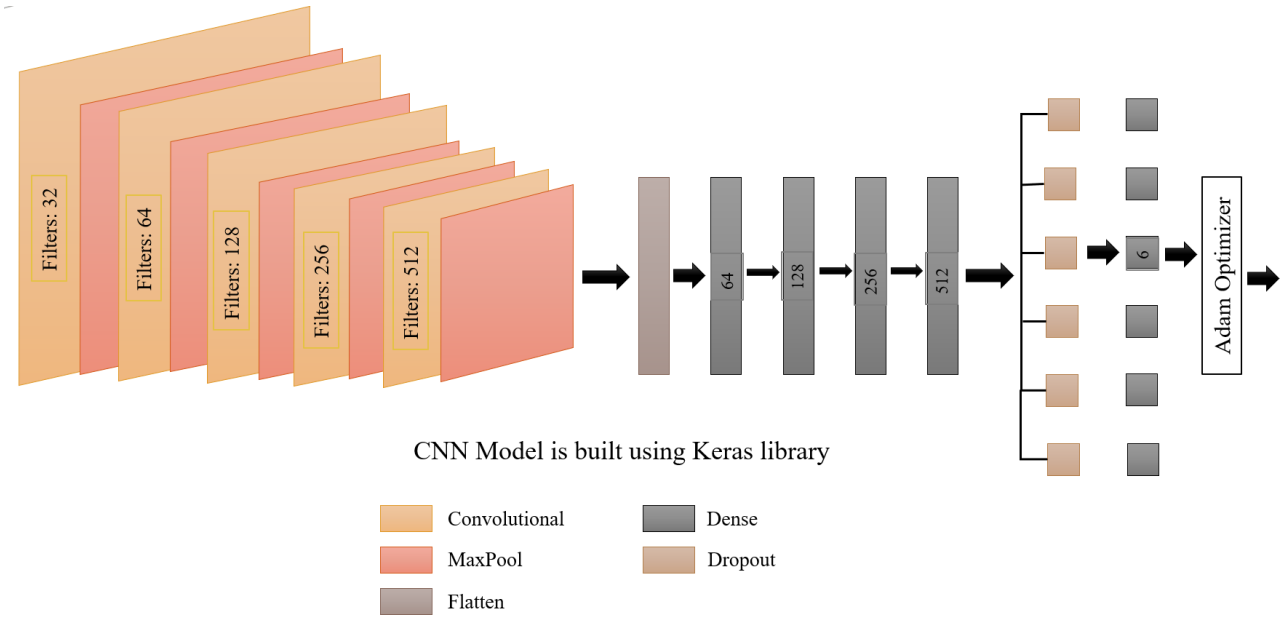

Fig 3: CNN Architecture

The model consists of 4 Conv2D layers of filters 32, 64, 128, 256 respectively. The kernel sizes specified are (3, 3) and strides are 2. The pool size is set to (2,2). As the first layer takes the input, input size is also to be specified. As previously seen that the images are of size 64 x 64, input size is set to (64, 64 ,3). The activation function used for each of these layers is ReLU i.e., Rectified Linear Activation.

Dense layers are also added. But before the dense layers, flatten layer is added to connect the convolutional layers and dense layers. After flatten layer, 6 dense layers are added. The units in the dense layer are 64, 128, 256, 512 respectively. All these layers use ReLU activation function. After these

layers, a dropout layer is added. Dropout layer prevents the model from overfitting. Dropout randomly sets the outgoing edges of hidden units to 0 at each update of the training phase. Dropout layer is passed a value of 0.2 i.e., every hidden unit i.e., neuron is set to 0 with a probability of 0.2. After dropout layer another dense layer is added. As this is the last layer, this can also be said as output layer. Since we have 6 different gestures to classify, thus there is possibility is 6 different outputs. As a result, the number of units in this layer is set to 6. The activation function used for this layer is SoftMax. It switches the result over completely to 1 so it very well may be deciphered as probabilities. The model will then, at that point, make its forecast in light of which choice has the highest probability.

The model is compiled where Adam optimizer is used with a learning rate of 0.001 and categorical_crossentropy as loss function with metrics set to accuracy. The Reduce learning rate function monitors val_loss with a factor of 0.2. The early stopping function monitors val_accuracy with patience set to 2. The number of epochs is set to 10.

The accuracy of the model is obtained to be 96.875% for training and 87.5% for testing.

Thus, the total parameters of the model are 630,022. The accuracy increases consistently for the first 4epochs and then remains steady.

2.4 Accessing mouse virtually

If hand is detected, then it is passed to CNN to identify gesture. If result from CNN is left, then left click is performed, if it is right then right click is performed. If the result is scroll_up then vertical scroll up is done for 15 clicks and is it is scroll_down the vertical scroll down is performed for 15 clicks. If the result is screenshot, then a screenshot is taken of current screen and saved with a .png extension. If it is palm, then cursor is moved to the normalized position of the base of the middle finger on the screen. A sleep time of about 0.5 seconds is introduced after left and right clicks so that the clicks are not overlapped and the user gets sufficient time to change the gesture. All these operations are done using the PyAutoGUI library of python. It is quite easy to use and performs all the operations accurately.

There are many other operations which can be added like horizontal scroll, but the working of PyAutoGUI function for horizontal scroll function i.e. hscroll() is limited. It is supported only by OS X and Linux platforms. Thus, this limits the number of operations that can be introduced.

3. RESULTS

This project has successfully implemented virtual mouse using convolutional neural network based on hand gesture recognition

After training, the CNN model has obtained the training accuracy of 96.875% and testing accuracy of 87.5%.

Table 1. Results

Result	Training	Testing
Accuracy	96.875%	87.5%
Val. Loss	0.2690	0.3697

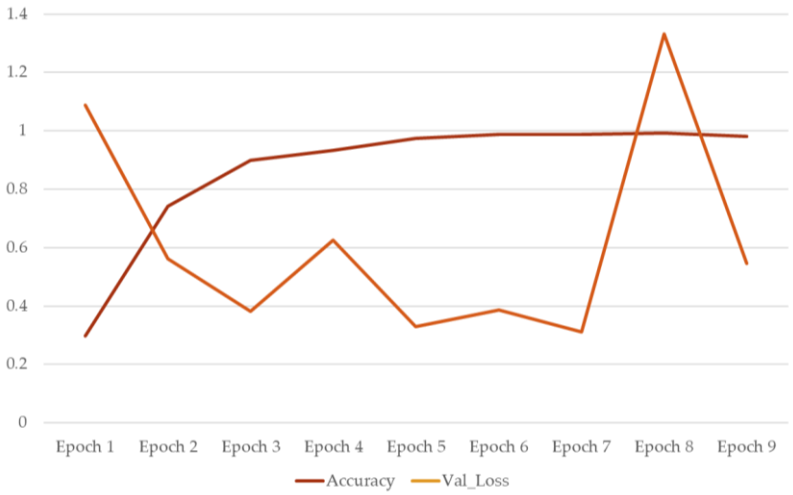

Fig 4: Model Output

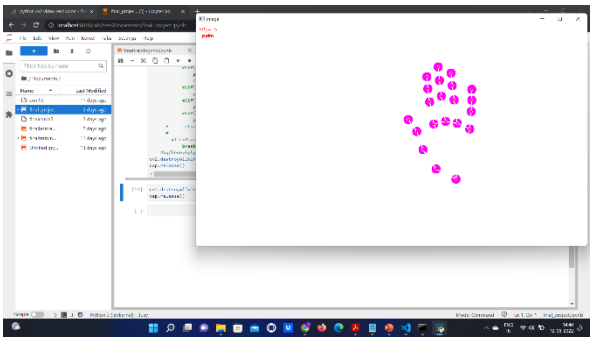

Fig 5: Recognition of Navigation Gesture

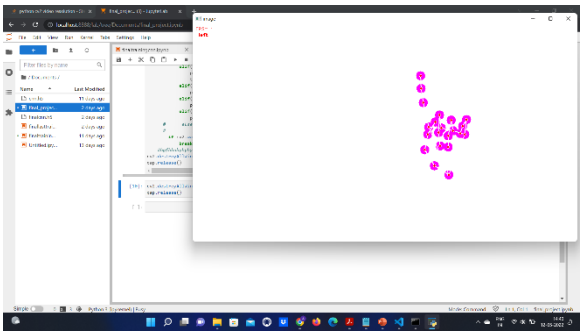

Fig 6: Recognition of Left Click Gesture

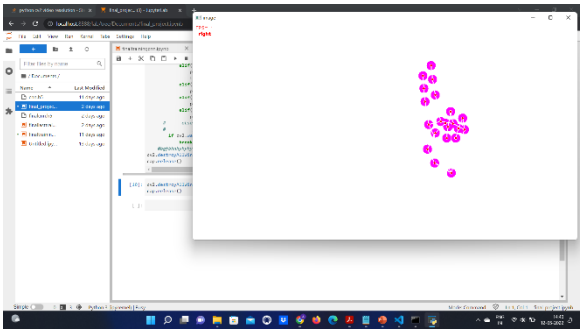

Fig 7: Recognition of Right Click Gesture

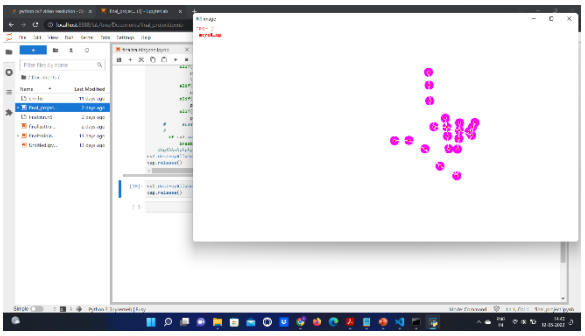

Fig 8: Recognition of Scroll Up Gesture

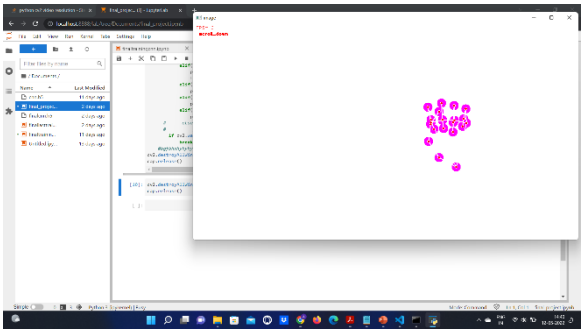

Fig 9: Recognition of Scroll Down Gesture

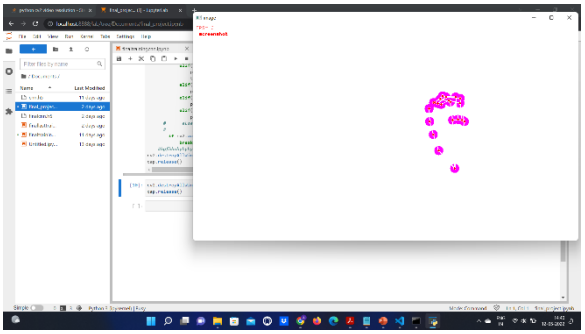

Fig 10: Recognition of Screenshot Gesture

4. COMPARISON WITH PREVIOUS PAPERS

Characteristics	Previous papers	This Paper
Pre-processing	Background subtraction method of OpenCV is used, weighted background estimation is used.	Array form of image is created where background is taken as white and 21 co-ordinates are marked with circles and joined with lines.
Algorithm used	Color identification and gesture recognition. Classified using area ratio calculated by convex hull.	Gesture Recognition. Classified using CNN.
Hand Gesture	Gesture is based on the number of fingers.	Natural gestures are used.
Accuracy	90%	87.5%
Additional hardware	Webcam Colored caps	Webcam

5. CONCLUSION

With technology advancing every day, improvements are made in each and every aspect. In this paper, we have developed a virtual mouse which is based on CNN. The user's hand is detected and the gestures of hand are identified by CNN and according to the gestures different operations are carried out. The detection of hand is not affected by the lighting conditions and background thus, overcoming the limitations of the previous systems. The developed system has quite high accuracy and is suitable for real-time use.

6. APPLICATIONS/ FUTURE WORKS

From future perspective, we can incorporate more mouse operations thus widening the scope of the project. By using different algorithms, we can also strive to increase the accuracy of the results. Apart from the virtual mouse system mentioned in the paper, this model can be combined with other technologies to provide support to differently-abled people. For e.g., sign language and communication. It can also be used in the gaming field. Further eye movements can be used to control the mouse which will be much more beneficial.

7. ACKNOLEDGEMENTS

Our thanks to our professors who encouraged and guided during the project.

Thanks to our colleges who helped in achieving this project.

8. REFERENCES

[1] Kumar, P. & Verma, J. & Prasad, Shitala, "Hand data glove: A wearable real-time device for human-computer interaction" International Journal of Advanced Science and Technology,2012.

[2] X. Zhang and S. Xu, "Research on Image Processing Technology of Computer Vision Algorithm" 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020, pp. 122-124.

[3] L. D. Singh, P. Das and N. Kar, "A pre-processing algorithm for faster convex hull computation" Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), 2013, pp. 413-418.

[4] Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Eu-Tteum Baek, Soo-Hyung Kim and Gueesang Lee "Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network" Appl. Sci. 2020, 10, 722.

[5] T. Palleja et al. "Simple and Robust Implementation of a Relative Virtual Mouse Controlled by Head Movements" Conference on Human System Interactions, 2008, pp. 221-224Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[6] V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition" 2020 IEEE-HYDCON, 2020, pp. 1-5.