Hand gesture-based recognition for interactive Human Computer using tenser-flow

Article · January 2020

CITATIONS
0

READS
254

4 authors:

Sangeeta Kumari
Vishwakarma Institute of Technology
17 PUBLICATIONS **163** CITATIONS
SEE PROFILE

Shubham Mathesul
Vishwakarma Institute of Technology
7 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Parth Shrivastav
Vishwakarma Institute of Technology
4 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Ayush Rambhad
Vishwakarma Institute of Technology
8 PUBLICATIONS **8** CITATIONS
SEE PROFILE

International Journal of Advanced Science and Technology
Vol. 29, No. 7, (2020), pp. 14186-14197

Hand gesture-based recognition for interactive Human Computer using tenser-flow

Sangeeta Kumari [1], Shubham Mathesul [2], Parth Shrivastav [3], Ayush Rambhad [4]
1,2,3,4, Computer Engineering Department, VIT Pune
[1]sangeeta.kumari@vit.edu, [2]shubham.mathesul19@vit.edu, [3]parth.shrivastav19@vit.edu, [4]ayush.rambhad19@vit.edu

Abstract

Today's modern era has been a witness of ever-increasing hand gestures-controlled computer, laptop and other electronics appliances-based applications. Image-based hand gesture recognition systems are very easy, simple and economical to use as these gesture-controlled applications does not involve any additional hardware. These applications mainly rely on inbuilt cameras. Such remote recognition is very unique and heavily influenced by its performance factors such as image processing, illumination conditions, objects interference, view of source input etc. Advantage of gesture recognition systems is that no physical contact is required between user and the gesture recognition-enabled device. Systems allows to interact with the interface and control devices using simple hand gestures. Also, gesture recognition technology saves crucial time. These systems are not only pleasant to implement in daily life but also helpful to help people having disabilities, these systems could also be packed together with IOT devices for relevant usage. Using gesture recognition system security and integrity could also be identified and boosted.

Index Terms—*Gesture Recognition, Tensor Flow, Keras, Recognition, Image, Processing, Gesture Controlled application, Gesture*

1. Introduction

Human interaction with Computer is a vision for finding solutions to various real-world applications such as object recognition to adapt and improve the interaction between computers and individual users in need. A gesture recognition device should fulfill the needs by analysing the user inputs. Gesture recognition has become important as it has tremendous scope of advancement, as human interaction with the computer is not just limited to use of keyboard and mouse but with the broad range of application development supporting virtual reality, security and many more. Applications supporting gesture recognition has become a notable part of computer whether it is related to iris recognition or face, advancement in gesture recognition is seamlessly developing.

Human gesture has the natural ability to depict human actions and feelings easily, thus extracting information from these vast data to identify a particular gesture proves successful. As data is extracted from gestures, system can be

trained to generate corresponding result which would potentially return a systematic approach of how to interact with the human based on the particular gesture provided by them. These virtual environments create a vast and unique data which makes the recognition of gesture more accurate as newer datasets get added into the recognition system.

We use gestures in our day to day activities. Even while we video communicate, we gesticulate more than talk, as human tendency, the significant use of gestures and context in it has wide variety which if processed can provide large data that can be used for ease in human communications which could be widely extended with applications through computer software.

Gesture refer to any non-verbal, communicated via body movement that is supposed to deliver a specific instruction or message. In the field of gesture recognition, a gesture is described as any physical action, large or small, that a motion sensor may perceive. In certain instances, the gesture

ISSN: 2005-4238 IJAST

meaning can often contain speech or vocal orders. Motion sensor perceives data received from camera as main source and interprets movements which is processed by proposed system.

Diverse forms of gestures may lead to effect on accuracy of data, gestures can be classified as

- Static gestures
- Dynamic gestures

A static gesture is a specific setup and position of the hand, depicted by a single picture.

A dynamic gesture is a moving gesture, represented by a sequence of various images [1].

Static gestures have higher accuracy rate as they have uniform properties whereas dynamic gestures comprise moving points, unstable pattern and unpredictable movements that need processing and reconstruction if needed which consumes more processing time. As dynamic gestures represent temporal sequence of pattern, it comprises velocity, shape, location, and orientation properties.

II. Literature Survey

Author Plouffe, Guillaume & Cretu, Ana-Maria[1] developed a static gesture based application for real time system which is based on contour detection. These Authors have used K-curvature algorithm for contour detection and tracking over fingertips and Dynamic time wrapping (DTW). Dynamic time wrapping (DTW) uses Euclidian distance to find shortest path of singer's curvature.

In paper [2] author Meenakshi Pawar proposed an algorithm for Hand gesture recognition method which is purely based on Image processing and they have not used any training model. The whole process of Gesture recognition starts with image enhancement and segmentation process. Firstly, RGB image is converted YCbCr and then image is converted into binary image on which segmentation is applied. After Segmentation vertical or horizontal orientation is determined and then centroid of finger is obtained. Finally Finger region is identified using finger tips.

Hanwen Huang[3], used a different algorithm which is based on skin tone after that contour detection and segmentation process is carried out followed by gesture recognition. Author Guillaume Devineau et al [4] introduces a CNN based hand gesture recognition using deep learning for Skelton data using parallel convolution methods. They have used two separate channels based on multi perceptron for feature extraction methods. They have carefully chosen GLOROT uniform initialization for training purpose and negative log-likelihood for loss estimation[9].

Author Ali A. Alani[5] et al also proposed hand gesture recognition system using CNN like authors [4],[9] and [13], but he uses adaptive CNN(ACNN) to detect hand gesture accurately regardless of hand size or hand position. To increase the accuracy before applying Adaptive CNN, the hand images are moved horizontal and vertical direction randomly up to 20%, which increases the size of data size images and this ACNN algorithm also helps in fine tuning of parameters.

Author Duhart and his team[6] deduced that the research and implementation of ANN had been a revolution in further improvement. It has given rise to Distributed Artificial Neural Controller (ANC) which can be easily used in distributed environment. Thus, we can build a global Artificial Neural Network which is composed of multiple ANC's and different mechanisms which can synchronize them. The results of testing ANN come to the conclusion that the learning rate increases significantly with the number of layers and neurons. Thus making it much faster and productive.

In [7], the authors performed a research on the flow of the hand gesture recognition and made use of the AdaBoost classifier based on the Haar feature extraction to extract the features for the gestures in a unsuitable environment. The authors also made use of the CamShift algorithm for tracking the hand gesture and identifying the hand gesture area in real-time[10].

In [8], the authors proposed a model which consisted of two parts, back-end and front-end. The back-end system consists of three modules: Camera module, Detection module and Interface module. The camera module was responsible for connecting and collecting data from different picture detector styles and transmitting this information to the frame detection module. The detection module was responsible for the image processing and finally the interface module is responsible for mapping the detected hand gestures to their associated actions.

In [9], the authors studied the Convolutional neural networks (CNNs) and stacked denoising autoencoders (SDAEs) for the recognition of 24 ASL (American Sign Language) hand gestures. They trained their model on the public databases on which recognition rates of 91.33 and 92.83 % were

ISSN: 2005-4238 IJAST

achieved. The authors made use of the rectified linear activations in the hidden layers of the network, to overcome the difficulties in the learning of CNN due to the increased depths.

Authors Dardas and Georganas [10], stated that the gesture detection is divided into three modules namely hand detection, tracking of hand and multiclass SVM. Here, multiple algorithm and techniques are also explained. For feature extraction they have used SIFT algorithm. K-means clustering is used to map unified dimensional bag-of-words vector, which is building block for SVM classifier. As per the results the accuracy rate is 96.23%.

In the paper of Nikam and Ambekar [11], convexity hull algorithm is used for detection of gesture. But the interesting part is instead of using RGB images which are by default captured by webcam, they are converted to HSV. So it helps to tackle with lightning condition requirements. Here multiple steps are performed to filter image and remove noise. As per the results the accuracy rate is above 95%.

Siddharth S. Rautaray [12], in 2012 found out that evaluations were not proper while studying 3D recognition patterns and proposed his study to preview existing systems and publish a detailed survey dependent on the standard set by organizations. He found out that appearance-based hand gesture representations were more prioritized than the 3D based gesture representations in the hand gesture recognition systems. As industrial applications required accurate advances and troubleshooting systems rather than error prone systems in the man to machine and machine to machine interactions, more interactive means of communication was prioritized done by hand gesture systems.

A. A. Alani and his team [5] in 2018 proposed Adapted Deep Convolutional Neural Network (ADCNN) which was adaptable for classifying hand gestures as static datasets which has varied properties like noise, light, granularity. Data augmentation was used for obtaining desired image from the real input image and consequently in turn increased dataset size. The augmented images that was produced for better recognition improved model training. ADCNN was enhanced by use of dropout regularization and L2 Regularization to overcome the difficulty of data overflowing and fitting and enhancing the training of model with best suitable input.

Hung-Yuan Chung [13], in 2019, productively combined the old image processing method alongside with the tracking method and the deep CNN that has been popular in recent years in hand gesture recognition research, achieving good recognition results given a reasonable computational load. The proposed overall hand gesture recognition system effectively combines three key components, namely, hand detection, hand tracking, and hand recognition. For hand detection, we use skin segmentation, noise processing, and background subtraction to detect the ROI of the first frame entering the webcam. For hand tracking, we use this ROI as the initial position of the tracking, and train the initial model by using the KCF algorithm.

This section provides insights into different methods that are used for hand gesture recognition. Here we can compare these methods with their benefits and drawbacks with their accuracy level. Till now there are 2 main things to be considered.

1.Hardware

2.Detection Methodology

Hardware –

The hardware that are used for gesture recognition are data gloves, hand belts and cameras. Data gloves uses different sensors to recognize the hand gestures, movements, etc. The sensors used in data glove are gyroscope for measuring the tilting of hand, flex sensor for tracking the movement made by fingers, accelerometer for measuring proper acceleration of hand, etc. The data glove is shown in Fig. 1. Data glove provide the highest accuracy among the hardware, but the disadvantage is cost and availability. It is very expensive, and it is not easily available. An explicit gear is to be ordered for it [10].

ISSN: 2005-4238 IJAST

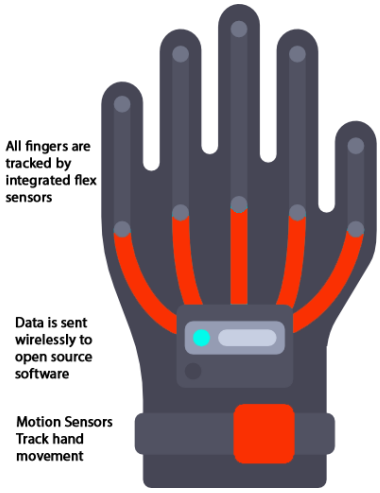


Fig. 1. Data Glove

The second hardware is the data belt. It is similar to data gloves and share the same advantages and disadvantages like it. But here the no of sensors that are attached to it are less than the data glove giving less accuracy. The third and most used is camera modules. Camera modules have no need to explicitly order, as most of the devices are already equipped with it. The camera modules cover most of the devices that ranges from laptop's webcam to phone's camera. One significant advantage is that, the user does not require wearing anything. Also, they are not expensive, but their accuracy rate is surely slightly less than other two. Here there is a trade-off between availability and accuracy. So, in most of the cases where we require high accuracy rate like gaming platforms, there we will go with data gloves. But for applications where the coverage of users is extensively high, there camera module has been proved as best [09].

To make our model cost-free and as our user coverage is high, we used an inbuilt webcam of laptop without the use of additional cameras or any other hardware. So, our primary focus in this paper would be related to the camera module.

Detection Methodology –

There are 3 main methodology. It includes image recognition techniques, template matching and neural networks. Here, in our model we have used a combination of two techniques, i.e. neural network and image recognition. These two techniques provide a higher accuracy rate, if the image is optimized enough. Thus, providing an efficient output. There are certain techniques like counting the convex hull in the image to recognize the gestures. In other image recognition technique, the frames captured by the webcam are cropped so that only some portion of it is rendered and processed. The steps for creating a small neural network is given in Fig. 2.

ISSN: 2005-4238 IJAST

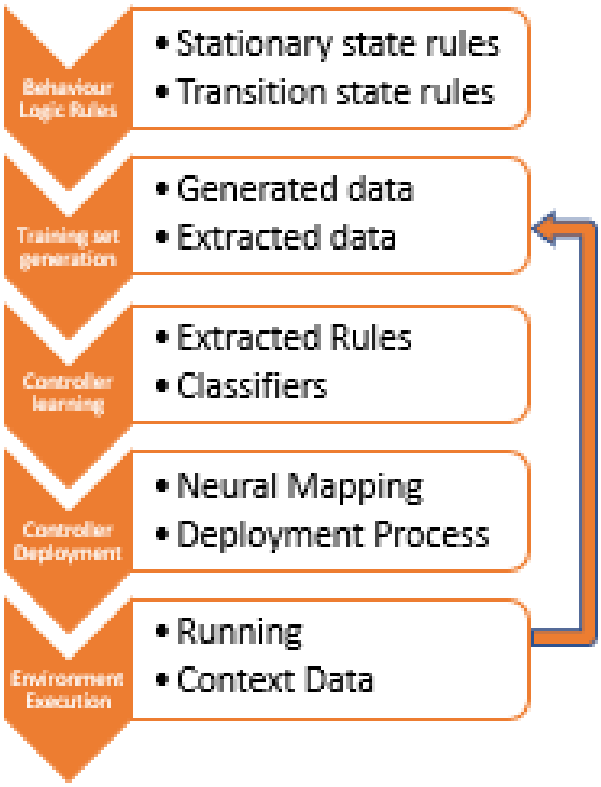


Fig. 2. Steps for creation of neural network

What you can tell from this, is that there are 5 major steps for building a neural network. First, we need to create behaviour logic rules. These are the rules which help in decidability. The data which fail these rules are filtered out in the next layers. The second step is training set generation. This data should be much precise as this has a major role for training your neural network. The third step is controller learning, there are multiple layers in this stage. It uses extracted rules which we made in first step. The classifier then applies these extracted rules for classification. The step 4 is Controller Deployment. It is mapping of neural layers with the classes generated. And last step is environment execution. This means running it with actual data. This step is looped with step 2 to test the neural network and to optimize it (i.e. removing illogical rules, etc)[9].

I. Architecture

The architecture of our system is shown in Fig. 2. It consists of three blocks that are as follows –

I. Frame Capture
II. Render and Processor
III. Classifier

ISSN: 2005-4238 IJAST

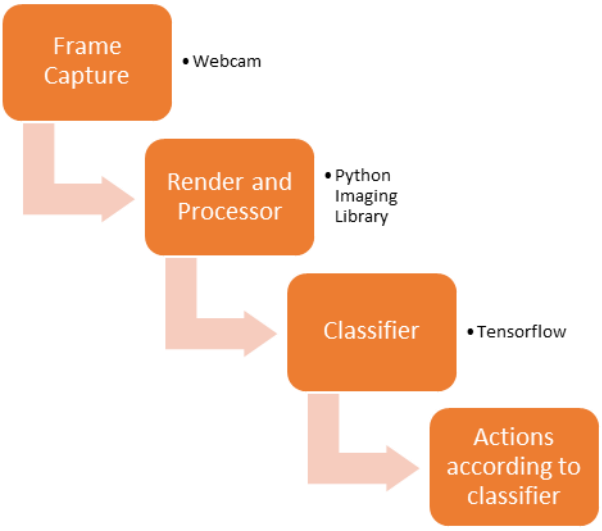


Fig. 3. Architecture of the Proposed System

The first block consists of frame capture. This step is done by using an inbuilt webcam of device. This doesn't need any more hardware other than that. Once the frame is captured it is sent for processing. This is done by python imaging library which is provided in python. Then it is sent to the classifier.

We used Keras model so that TensorFlow can directly predict and classify the captured image. Here we used the process of transfer learning. As traditional machine learning does not retain or accumulate its knowledge, thus it is never reused. On the contrary, transfer learning reuses the knowledge of previous models. Fig. 3 shows that how transfer learning is effective than training from scratch. Transfer learning not only improves accuracy but also saves time. The fig. 4 shows that the classification of car and truck is actually based on pretrained convectional neural network model which was used for classification of cats and dogs.

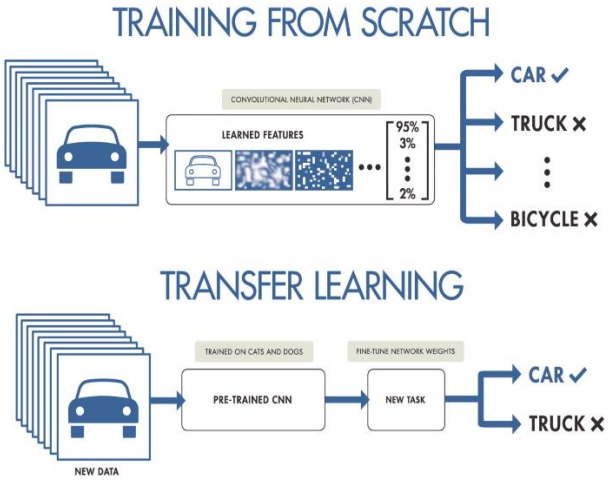


Fig. 4. Transfer learning

The last block states that, appropriate actions will be performed according to the prediction which has the highest value.

III.Working of Proposed System

In this section, in detail explanation of architecture has been provided. For our proposed system, we have used python as front end, as well as back end language. We choose python, for its simplicity and availability of libraries when compared to other programming languages. For capturing frames of images, we used a library named cv2 which is part of OpenCV. OpenCV is an open source

ISSN: 2005-4238 IJAST

library which is used for real-time processing. In our proposed system we have used this library for real-time image processing. We used its in-built methods for capturing as well as resizing it [2][3].

Python imaging library is used to reduce render and process the image. This is because OpenCV does not provide many features related to advanced processing and rendering.

We used Google's teachable machine project for training machine learning model to classify different hand gestures. For this we first provided dataset for different hand gestures. Different classes were generated for each hand gesture implemented. After all classes have sufficient data, we trained the data by setting appropriate values for epochs, batch size and learning rate. Thus, after training the model we can test the model and export it in Keras model. We have specifically exported model in Keras h5 format. There are two main types of Keras models. They are sequential model, and the model class which is used with functional API. We have used model class for loading and prediction purposes. The image is then used as an input to our classifier. For classifying, we have used TensorFlow. TensorFlow provides multiple functionalities while working with Keras model. After loading the model, we need to convert the image into NumPy array for preparing it for classification. Then we can predict the model by using predict() method. Fig. 5 shows about the layering filters of the input image and gives different prediction values [4].

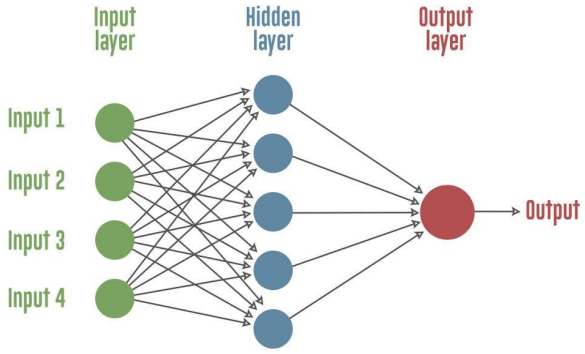


Fig. 5. Layering of CNN

Multiple input classes are filtered from multiple hidden layers. Once they are filtered every class get a prediction value which ranges from 0 to 1. The number greater than 0.9 is considered as a match.

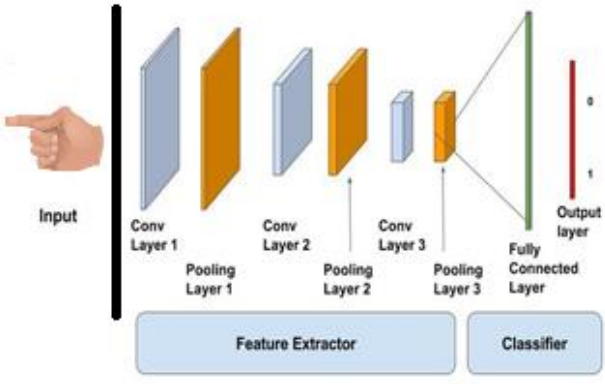


Fig. 6. Detailed CNN layering structure

As the fig. 6 shows, CNN consists of two main parts i.e. Feature Extractor and Classifier. Feature Extractor consists of pair of conventional layer and pooling layer. As you can see both the layers apply different behavioural rules to filter out the images. This data is used by the classifier, It is fully connected layer which classifies the input image and give the predicted value ranging from 0 to 1 [5].

ISSN: 2005-4238 IJAST

For implementing different actions after getting the prediction, we have used a library named PyAutoGUI. It lets python scripts to control and automate different interactions made with keyboard or mouse. It has methods to press keys on keyboard. We can not only just tap the keys but also, we can use different shortcut keys.

We have also used speech recognition in our proposed system, to perform speech-based tasks as well. For performing this we used speech recognition library which uses google speech recognition which is used to perform certain actions that are mapped with voice. Thus, making proposed system user friendly and interactive [6][7].

IV. Implementation

With the implantation of TensorFlow Framework and Keras model for deep learning and training the model, gesture recognition accuracy has inevitably increased. As a new image get inputted into the system, it is compared with the already trained model and processed thereafter.

Once gesture data are processed, matched and evaluated, if the gesture is recognized and properly classified by the model, it carries out the respective operation by issuing back response to the gesture system.

We used the hand gesture recognition database that Kaggle provided for our project. It includes 20000 pictures of different hands and hand gestures. The data collection includes a minimum of 10 hand movements involving 10 separate individuals. We have five female subjects and five male subjects.

Firstly, we divided our data into a training set and a test set. Our model will be built from the training set. The test data will then be used to check whether our projections are correct. Then, we made use of a seed that helped us maintain the randomness of our results which can be reproduced. The activation function takes the values present in the image which are just a sequence of numbers and increases their non-linearity which is essential as the images are of non-linear form. The most common activation used for achieving this is the Rectified Linear Unit (ReLU).

Once we had the model created and ready to be validated, we set some parameters to ensure maximum accuracy from the developed model. Some of the parameters included a dropout layer which was used to avoid overfitting, a batch normalization function to make sure that the inputs were normalized while moving into the next layer, and finally the softmax activation function chooses the neuron with the greatest likelihood, emphasizing that the gesture belongs to that specific class.

We trained our model on 16,000 samples while using 6000 samples out of them for validation purpose. The number of epochs for training the model were 50, with each having a batch size of 64. The test accuracy on validating the sets after all the epochs resulted in being 0.9995 which meant an accuracy of 99.95% was produced by our model.

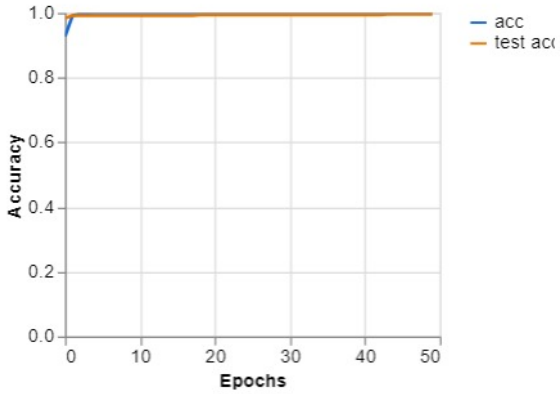


Fig. 7. Average accuracy per epoch.

ISSN: 2005-4238 IJAST

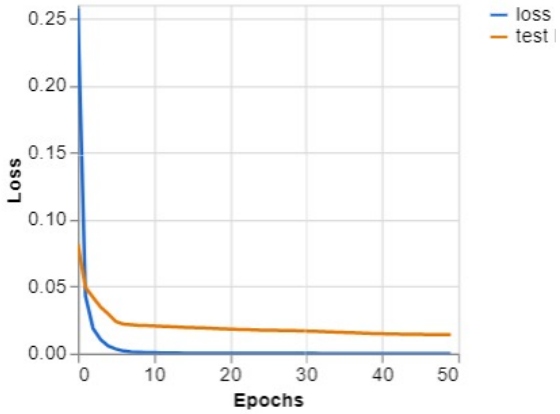


Fig. 8. Average loss per epoch.

Figure 7 and 8 provides a brief overview of the results achieved in terms of the average accuracy and loss per epoch of our model. The table below contains the confusion matrix comparing the performance of our model between the predicted and actual results on the set of data which was provided for validation.

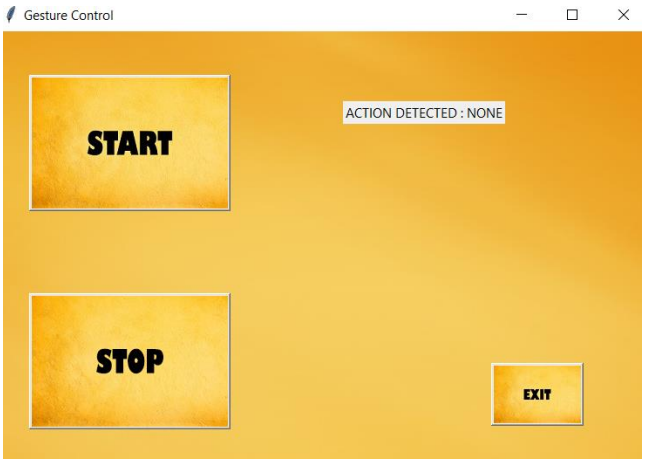
	Predicted Left	**Predicted Right**	**Predicted Palm**	**Predicted Fist**
Actual Left	**625**	**0**	**0**	**0**
Actual Right	**0**	**607**	**0**	**0**
Actual Palm	**0**	**0**	**600**	**0**
Actual Fist	**0**	**0**	**0**	**585**

Table 1. Confusion matrix indicating the predicted output against the actual output.

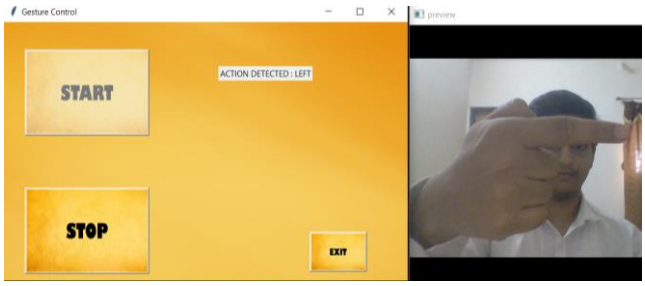
V. Experiment Setup and Result Analysis

GUI:

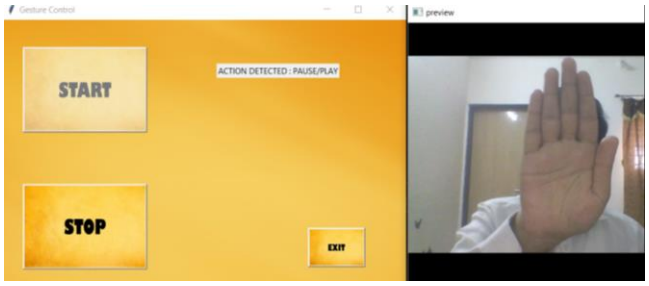
ISSN: 2005-4238 IJAST



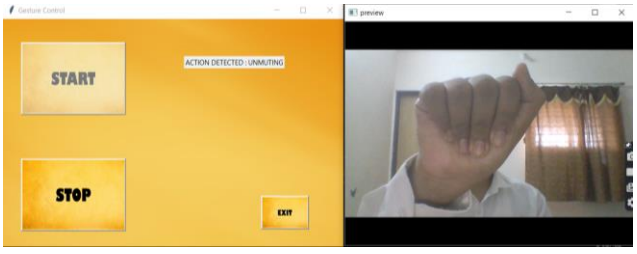
As soon as you run our software program, the toolbar appears in the middle of the computer. Our toolbar contains several buttons which allow the user to choose between our system's audio model and video model.



Clicking on the start button begins the video model recording your movements using your computer's integrated webcam.



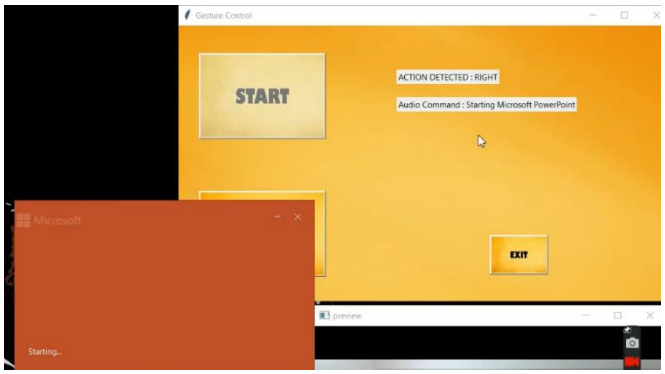
Whenever you make a particular movement to the integrated camera of the computer, it checks the motion and informs you the recommended action to be taken and performs that specific action on the computer. The specific action in the picture performs the ‘Play / Pause’ action on your playback.



The specific action in the above picture performs the ‘Mute / Unmute’ action on your playback.

ISSN: 2005-4238 IJAST

You may attach a microphone or specifically say the activity to be done in front of the computer screen, our software program analyses the audio and then tells you of the suggested action to be taken and carries out the particular action on your computer. The specific action in the picture performs opening the 'Paint' application program on your computer system.



The specific action in the picture performs opening the 'Microsoft PowerPoint' application program on your computer system.

If the application program you have asked to run is not available on your computer device, our software system would open the browser window and then guide you to the application's installation page so that you can download it and install it on your computer system to use it.

VI. Conclusion

This paper proposes a Gesture recognition system that manipulates hand gesture both static and dynamic according to their properties. Accuracy of this system is 94.44% as datasets used to train model included various unique layers of images using which the system efficiency and accuracy has improved.

This gesture recognition system proved to be useful for not only while using desktop or laptop applications but also in IOT based devices that includes smart TV's, and many more.

Usability of this system is high because of its simplified design and robust processing which provides higher processing in less computation time.

This gesture recognition system not only gravities technological development but also eases human interfacing with computer, in case of person with disabilities it eases accessibility and makes particular individual independent of themselves for performing any activity.

REFERENCES

[1] Plouffe, Guillaume & Cretu, Ana-Maria. (2015), " Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping," IEEE Transactions on Instrumentation and Measurement. 10.1109/TIM.2015.2498560.

[2] Panwar, Meenakshi & Mehra, Pawan. (2011), "Hand gesture recognition for human computer interaction," 1-7. 10.1109/ICIIP.2011.6108940.

ISSN: 2005-4238 IJAST

[3] Hanwen Huang et al, “Hand Gesture Recognition with Skin Detection and Deep Learning Method”, J. Phys.: Conf. Ser. 1213 022001
[4] Guillaume Devineau, Wang Xi, Fabien Moutarde,Jie Yang, “Deep Learning for Hand Gesture Recognition on Skeletal Data,” 13th IEEE Conference on Automatic Face and Gesture Recognition (FG’2018), May 2018, Xi’An, China. 10.1109/FG.2018.00025. hal-01737771
[5] A. A. Alani, G. Cosma, A. Taherkhani and T. M. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," 2018 4th International Conference on Information Management (ICIM), Oxford, 2018, pp. 5-12. doi: 10.1109/INFOMAN.2018.8392660
[6] Duhart, Clément & Bertelle, Cyrille & Paris, Ece & Lacsc, France & Fr,. (2014), “Methodology for Artificial Neural Controllers on Wireless Sensor Network,” 10.1109/ICWISE.2014.7042663.
[7] J. Sun, T. Ji, S. Zhang, J. Yang and G. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 2018, pp. 1-4.
[8] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar , J. S. Nayak, “Hand Gesture Recognition for Human Computer Interaction ”, 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22- 24 August 2017, Cochin, India.
[9] Atul B Kathole, Dr.Dinesh N.Chaudhari, “Pros & Cons of Machine learning and Security Methods ”, 2019.http://gujaratresearchsociety.in/index.php/ JGRS, ISSN: 0374-8588 ,Volume 21 Issue 4
[10] Atul B Kathole, Dr.Prasad S Halgaonkar, Ashvini Nikhade, “ Machine Learning & its Classification Techniques ”,International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-9S3, July 2019.
[11] O. K. Oyedotun, A. Khashman, “Deep learning in vision-based static hand gesture recognition”, The Natural Computing Applications Forum 2016.
[12] N. H. Dardas and N. D. Georganas, "Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques," in IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 11, pp. 3592-3607, Nov. 2011, doi: 10.1109/TIM.2011.2161140.
[13] A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786.
[14] Rautaray, S.S., Agrawal, A. Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43, 1–54 (2015). https://doi.org/10.1007/s10462-012-9356-9.
[15] H. Chung, Y. Chung and W. Tsai, "An Efficient Hand Gesture Recognition System Based on Deep CNN," 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, Australia, 2019, pp. 853-858, doi: 10.1109/ICIT.2019.8755038.

ISSN: 2005-4238 IJAST