International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Hand Gesture Recognition to Perform System Operations

Anirudh Poroorkara[1], Dhiren Pamnani[2], Aayush Makharia[3], Dr. Madhuri Rao[4]

[1, 2, 3] *U.G. Students, Department of Information Technology, TSEC College, Mumbai, Maharashtra, India*
[4]*Professor, Department of Information Technology, TSEC College, Mumbai, Maharashtra, India*

Abstract- *Hand gestures have been the mode of communication for humans before we learned to speak and communicate verbally. It is the most convenient form of communication for two individuals to interact with each other without the barrier of language. In this paper, we introduce a more efficient way of utilizing hand gestures for operating systems. The data model uses Deep learning, Convolutional Neural Networks and a dataset from Kaggle with more than 50,000 training and testing images. The program then uses an algorithm to recognise the background, segments the hand gesture and then recognises the gesture. This paper tries to analyse the performance of the program under different backgrounds and the accuracy of the trained model.*

Keywords: Deep Learning, Convolutional Neural Network, Python, PyAutoGUI, Problem-solving, Computer Science

1. INTRODUCTION

From the start of the 21st Century, we all have aspired to control any device with just a movement of our hand or wrist. Our hands have always played an important role in helping us learn and remember. Therefore, Hand Gesture Recognition is a perceptual computing user interface that allows devices to capture and interpret these gestures as commands. These devices can then execute commands based on these unique gestures.

Hand Gesture recognition works by providing real-time data to a computer and discards the need of traditional data input like typing or using a pointer. It only requires a standard input camera that feeds the image into a software running a data model. The data model can recognise meaningful gestures from a list of defined and trained gestures. To train the data model with the data images, Deep Learning with Convolutional Neural Networks has been used. Convolution Neural Networks extracts relevant features by analysing the images and is extremely fast and efficient. We use a data set of size 53,620 images, and therefore Deep learning is used to outperform other learning techniques. Finally, the model uses Adam to optimise our deep learning algorithm.

The primary issue faced in gesture recognition is to find a stable background so that the model can clearly find the Hand. We use an algorithm from OpenCV called accumulated Weighted () to find the running average over a frame sequence. This enables us to find the difference between the backgrounds and foreground so that the latter can be distinctly identified. This paper discusses implemented model of gesture recognition that recognises static gestures only.

2. RELATED WORK

In a model proposed by Shivendra Singh [1] uses Hue, Saturation and Intensity values of skin to detect the hand and pre-processing is applied to remove the non-hand region. The region in the image which represents hand is set to 0 and rest is set to 1. The image is converted to binary image format. City Block Distance transform method is used to detect the centre point of the palm. Finger tips are detected by dividing the image in two vertical halves and scanning each half from top to bottom to find the highest point which are the finger tips. Gesture classifier is then used to classify the gesture which is based on the number of finger tips detected and the angle between fingertips and the palm point. The performance of the proposed work in this paper is highly dependent on the environment where it is trained and tested as it requires a background which does not have HSI values similar to skin.

Simran Shah [2] proposed a model using Convolution Neural Network to classify gesture based on the training data. A CNN model was built with two convolution layers followed by a pooling layer with an output layer of 7 nodes as the model was trained to accurately identify between 7 gestures. The images were pre-processed with two modes Binary mode and SkinMask mode. In binary mode the images were converted to grayscale and Gaussian blur was applied to smoothen the image and remove noise. In SkinMask mode the image is converted to HSV format and HSV range for skin is applied to extract only the region with hand.

A model to recognise American Sign language was proposed by Rutuja J [3]. The CNN classifier takes input image and process it through its convolution layers to extract specific features and then is passed through a fully connected ANN to classify gestures based on the features extracted. It contains two parts, assign language to text converter and a text to Sign language converter.

3. PROPOSED MODEL

This model is can be divided into four sections – pre-processing, data model, real-time running of the model and recognition of the gesture.

3.1 Pre - processing

For training the model, we have used Multimodal Hand Gesture dataset from Kaggle [4]. It contains 19 different gestures over 53,620 images taken in grayscale format. The images from the dataset are first resized to dimensions 56x56 for making learning easier and converted to NumPy array of type float32. Gaussian Blur of 5x5 is then applied to the NumPy array to smoothen the edges. Finally, binary threshold function with a threshold value of 30 is applied to distinctly find the edges and shape of the hand from the background. The dataset is divided into training set consisting of 42,808 images and test set consisting of 10,812 images

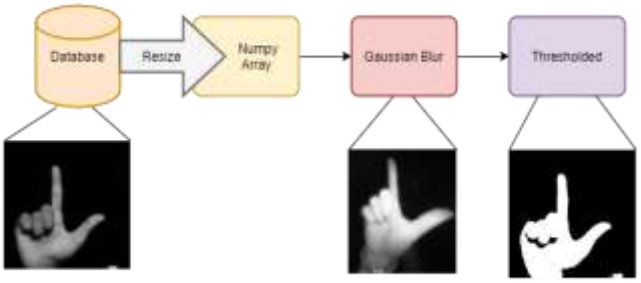

Fig. 1. Pre-processing the image in dataset

3.2 Deep Learning with Convolutional Neural Networks

The model contains 3 convolution and max-pooling layers along with Artificial Neural Networks. These layers are used so that specific features placed in the image can be extracted without any constraint as to where they are placed. The convolution layer is used to preserve relationships between pixels by using filters of size n x n and is used for feature extraction. Pooling layer is used to reduce the number of parameters after convolution [5].

The first convolution layer contains 32 filters of size 5 x 5. It is responsible for learning low level features from the image. The next two convolution layers contain 64 filters of size 3 x 3. Rectifier Linear Unit (ReLU) is used as activation function for each of the convolution layers. ReLU returns linearly for positive values and zero for all other values.

The output of the last pooling layer is in matrix form containing all extracted features. The Flatten function is used to transform the two-dimensional matrix of features from the convolution layer into a single vector to be fed to the dense layers. These values are placed in a single vector so that it can be provided as input for the Artificial Neural Network. We use two dense fully connected layers, the first layer consisting of 258 nodes and the second layer consisting of 128 nodes. The dense layer is used to generate the prediction from the feature maps provided by the input vectors. Both of these layers also use ReLU as activation. The last output layer is regular neural network layer which receives input from the previous dense layer and computes the class scores. This final layer uses SoftMax function as activation. SoftMax is used when an instance is to be assigned to one class when the number of classes is more than 2. It then outputs the 1-D array of size equal to the number of classes along with its probabilistic values. The output layer of this model contains 19 nodes, one for each gesture.

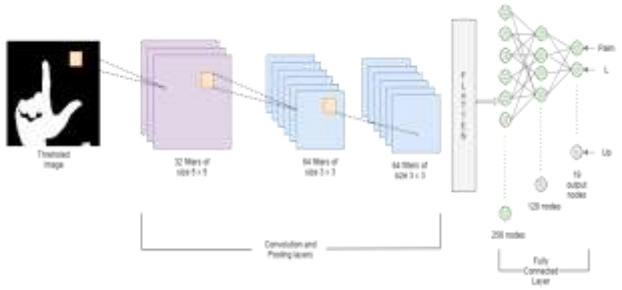

Fig. 2. CNN architecture

3.3 Real-time Capture

As soon as the program runs, the background is calibrated. The first 100 frames are to be kept free of any movable object and is treated as the background. This is done using a function called Running Average from OpenCV [6] which creates a background using the given 100 frames and stores it in a variable. The function employees a method called cv2.accumulatedWeighted() [7]. Any object introduced into the frame here after will be treated as a foreground image. Region of Interest is defined and is converted to grayscale before processing.

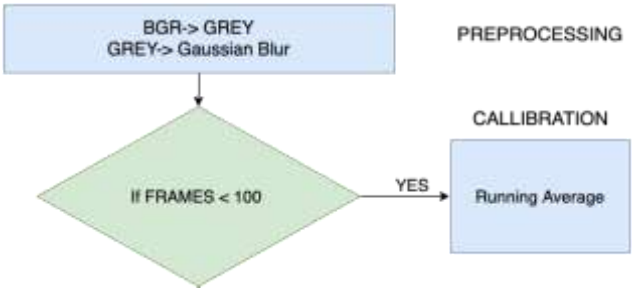

Fig. 3. Background calibration

3.4 Segmenting and Recognition

In the Segment function, the absolute difference between the input frame and the background frame is found. This blackens the background and any object lighter than the background is whitened. Gaussian Blur and the same Binary Threshold function are then applied on the image. The purpose of this whole function is to distinctly find the "hand" from the image and send it to recognition.

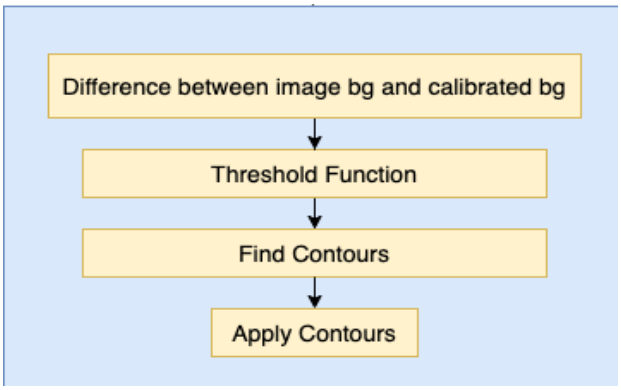

Fig. 4. Pre-processing of real time image

The Recognition functions converts the image to NumPy array, reshapes it and then sends it to the model for prediction. The model gives a probabilistic value for each node and the sum total of all nodes will always be 1. The node with the highest probabilistic value is the result of the convolution neural network and is displayed in the frame

4. RESULT AND ANALYSIS

The deep learning model gives an accuracy of 86.39% on training with arguments 20 epochs and a batch size of 64. Adam optimiser [9] gives us huge performance gains in terms of speed of training and is instrumental in training of the deep learning model. The real time algorithm eliminates background and only sends the foreground image of the hand. After comprehensive testing, we find that out of the trained 19 gestures, the setup is only able to recognise 7 gestures accurately. There are also some additional constraints that need to be assumed for the gesture to be recognised accurately. The background of region of interest, once calibrated cannot be changed. This will cause the background to distort which will give us unsatisfactory results. However, the background elimination method using Running Average method, seems to be more effective in finding contours and the distinct shape of the hand.

Fig. 5. Implemented program recognising Palm gesture

5. CONCLUSION

Thus, the proposed model introduces a gesture classification system that can identify 7 different static hand gestures. Each hand gesture image is pre-processed through a series of steps. The image is resized and converted to grayscale. This image is then converted to NumPy array and sent to the model for training. The trained model is then used to predict the real time hand gestures.

The program uses the initial frames to determine the background using the Running Average method. When the hand is introduced in region of interest, it uses Segment method to eliminate the background and only take the hand as input. This way the system can classify the gesture but can easily detect the gesture in any stable environment. The background elimination also helps to increase the model's accuracy against disturbances.

6. FUTURE SCOPE

Hand gesture recognition system has a variety of use as it helps to interact with the computer without having the need to manually touch the machine. It provides a way of remote interaction. This hand gesture recognition system provides with a new efficient and faster way of interacting with the computer and will soon replace the need to use mouse or keyboard for quicker operations. One of the most valuable ways of using this technology is to use a framework to control the system functions using hand gestures as input. PyAutoGUI is a cross-platform GUI automation Python module which can be used to control mouse and keyboard functions. The system can be used as an input method and each hand gestures can be assigned specific tasks of mouse and keyboard. PyAutoGUI provides the facility to programmatically control the mouse and keyboard. Therefore, when a specific gesture is detected by system, the corresponding task assigned to it is invoked and performed. Thus, this way the hand gesture recognition system can be used to control computer operations.

REFERENCES

[1] Shivendra Singh, Real Time Hand Gesture Recognition Using Finger Tips, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 06 | June -2018 page 1420

[2] Simran Shah, A Vision Based Hand Gesture Recognition System using Convolutional Neural Networks, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 04 | Apr 2019 page 2570

[3] Rutuja J, Hand Gesture Recognition System Using Convolutional Neural Networks, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 04 | Apr 2019 page 4508

[4] Multimodal Hand Gesture dataset https://www.kaggle.com/c/multi-modal-gesture-recognition

[5] Tasneem Gorach, DEEP CONVOLUTIONAL NEURAL NETWORKS- A REVIEW, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 07 | July-2018 page 439

[6] OpenCV. Open Source Computer Vision Library. 2015, https://opencv.org/

[7] https://docs.opencv.org/2.4/modules/imgproc/doc/motion_analysis_and_object_tracking.html

[8] Rafiqul Zaman Khan, COMPARATIVE STUDY OF HAND GESTURE RECOGNITION SYSTEM, Department of Computer Science, A.M.U., Aligarh, India

[9] Diederik P. Kingma, Jimmy Lei Ba, ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, Published as a conference paper at ICLR 2015