HAND GESTURE RECOGNITION SYSTEM USING KERAS

MR. KRISHNA BIHARI DUBEY, MOIN MALIK , TAUFEEK MANSURI

Department of Computer Science & Engineering

ABES INSTITUTE OF TECHNOLOGY, GHAZIABAD

Abstract - Hand gestures are an effective form of communication for everyone. Hand gesture recognition can be seen as a way for computers to better understand human body language, thus building a richer relationship between humans and computers rather than the traditional text based or GUI based approach.
Hand gestures can make tasks simpler by giving commands to the computer with simple movements of the hands. The gestures are captured by the camera, interpreted by the computer and appropriate action is taken in response to the gesture This project is based on hand gesture recognition to capture the hand gestures and convert them to commands understand- able by the machine and perform appropriate tasks. In this project we firstly capture the gesture using a camera. The captured image then undergoes pre-processing for feature extraction and implement convolutional neural network using Back Propagation Algorithm to effectively recognize the gestures and train the machine.

Keywords - Hand Gestures, Image Processing, Deep Learning

I. INTRODUCTION

Gestures are a potent means to express feelings, thoughts, and words. With the advancement in computer vision and artificial intelligence it is now possible for computers to capture and identify human hand gestures thus abridging the gap in human machine interaction.
Hand gestures can be effectively used in giving commands to your computer. The computer will capture the gesture, interpret it and take the appropriate action based on its previous inputs and data. This makes our day to day tasks easy and enhance the human computer interaction over the regular text based commands and GUI.
Human computer interaction takes place mainly by means of input output devices such as mouse, keyboard, joystick etc. we type commands from the keyboard or select an option using the mouse. These tasks can be simplified is these traditional input output mechanisms are replaced by hand gestures. Using hand gestures you can ask computer to perform various tasks like if you want to open a particular application in your computer whole name starts with L, if you make a gesture indicating the letter L with your hand and your computer will show you all the applications whose name starts with the letter L, then you can select your desired application.
Hand gesture recognition system has wide application in today’s world. With the advent in computer vision we can make our machine perceive knowledge of the surroundings just like humans. Today computers can inter- pret a gesture just like humans can. This helps in providing ease and naturalness for human computer interaction

II. TECHNOLOGIES USED

OpenCV [1]
OpenCV stands for Open Source Computer Vision Library .It has C++, Python and Java interfaces and supports many operating systems like Windows, Linux, Mac OS, iOS and Android. OpenCV is designed to improve compu- tational efficiency and provide a strong focus on real-time applications.

Python [2]
Python is an interpreted, high-level, general-purpose programming language.

PIL (Python Imaging Library) [3]
Python Imaging Library which is also known as Pillow in newer versions is a free library for Python. It provides support for displaying and manipulating, different image file formats.

Numpy [4]
NumPy is a library used by the Python programming language. It provides support for large, multi-dimensional arrays and matrices, and also has a large collection of high- level mathematical functions to operate on these arrays.

Tensorflow [5]
TensorFlow is an open source software library which provides high performance numerical computation. It pro- vides strong support for machine learning and deep learn- ing and the flexible numerical computation core is used in many other scientific domains.

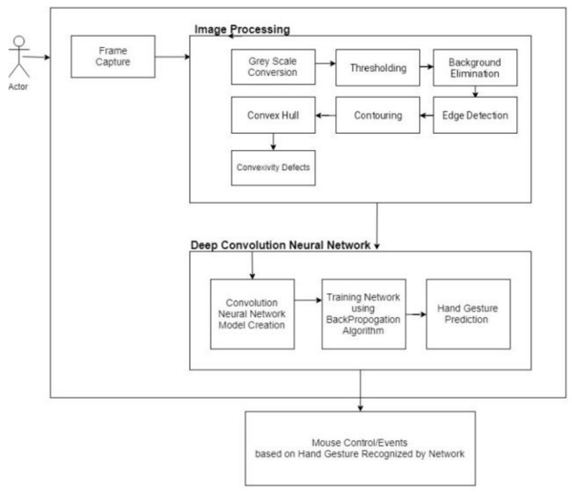
Keras [6]
It is an open-source neural-network library written in Python language. It is capable of running on top of Ten- sorFlow and it has many implementations of commonly used neural-network building blocks such as layers, ob- jectives, activation functions, optimizer’s, and many tools to simplify working with image and text data.

III. IMPLEMENTATION

In this system we make use convolutional neural network using back propagation algorithm to classify 9 hand gestures namely Swing for double click, index finger pointing up for single click, fist for dynamic movement, acknowledge for fine movement, thumbs up for saving a document, gun for a particular application for example game to open.

Gray scale conversion

The image is converted to grayscale for processing.



Actor
Frame Capture
Image Processing
Grey Scale Conversion
Thresholding
Background Elimination
Convex Hull
Contouring
Edge Detection
Convexivity Defects
Deep Convolution Neural Network
Convolution Neural Network Model Creation
Training Network using BackPropogation Algorithm
Hand Gesture Prediction
Mouse Control/Events based on Hand Gesture Recognized by Network

Figure 1: System Architecture

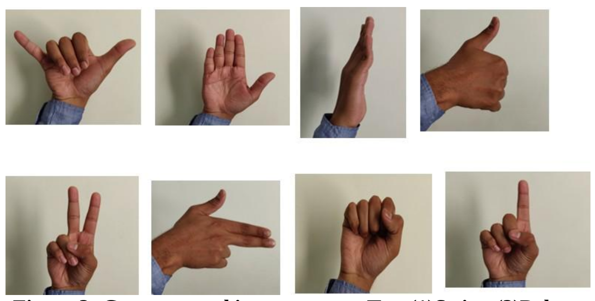


Figure 2: Gestures used in our system. Top (1)Swing (2)Palm (3)Straight (4)Thumbs up. Bottom (1)double finger (2)Gun (3)Fist (4)Single finger

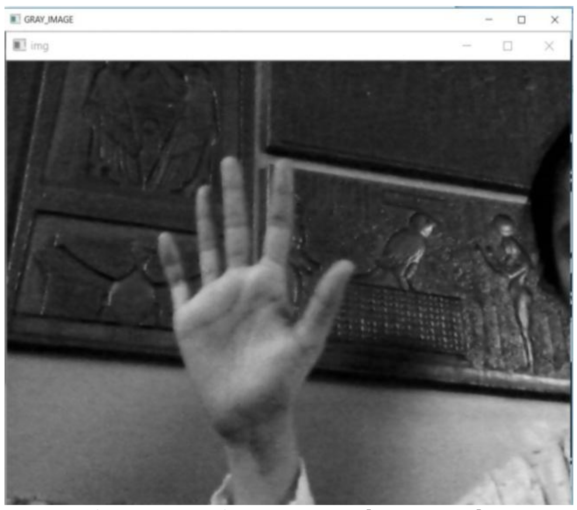


Figure 3: RGB image converted to Grayscale

Thresholding

Thresholding replaces each pixel in an image with a black pixel if the image intensity is less than some fixed constant T(specified threshold value), or a white pixel if the image intensity is greater than that threshold value.

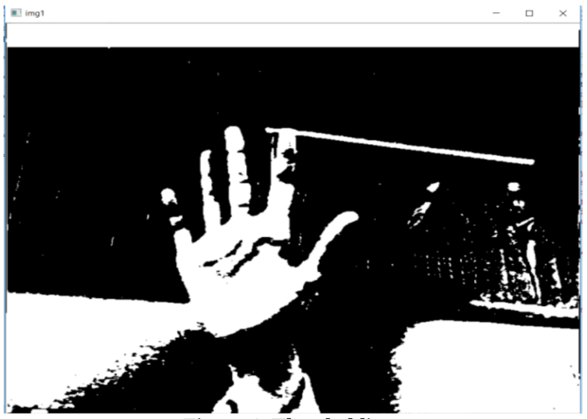


Figure 4: Thresholding

Background elimination [7]

There is a lot of unwanted background in the surrounding. This background is not required and could create con- fusion while identifying the hand gesture. Thus, we need to eliminate the background by running averages method.

In this method we create a model of the background which remains static. We create a mask by capturing the dynamic gesture in the current frame and subtract this frame from the background model. Running averages method can be implemented using the acculu- mateWeighted(input image, output, alpha) function in. al- pha denotes the weight of the input image. alpha controls the update speed i.e how fast the accumulator "forgets" earlier images. It means, if alpha has a higher value, aver- age image tries to capture even very fast and short changes in the surrounding. If alpha has a lower value, average image becomes slow-moving and it won't consider fast changes in the input images. This function helps us to create the background model which we will subtract from the current frame. This will give us only the hand region without the unwanted background.

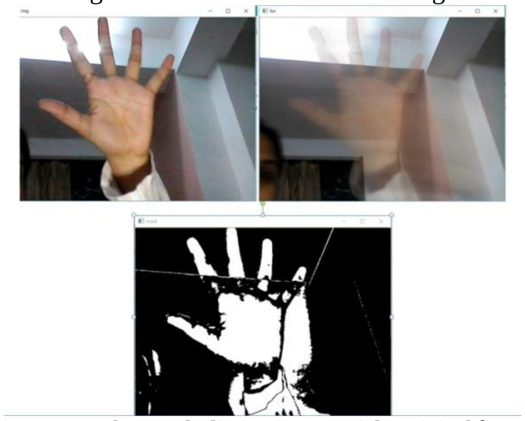


Figure 5: Background Elimination Top right-original frame Top left-static background model Bottom - mask obtained by difference of background model and original frame

Edge detection

Once thresholding is performed, we obtain the edges of the hand region using canny edge detection algorithm. The edges are found by finding the points where there are large discontinuities or sharp changes in brightness values of pixels.

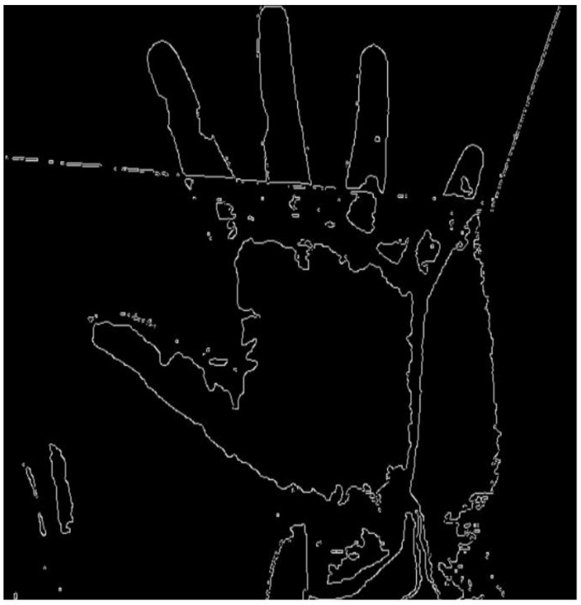


Figure 6: Edges of the hand using Canny edge detection algorithm

Contouring

Contours are straight lines or curves describing the in- tersection of one or more horizontal planes with real or hy- pothetical surface with. The contour is drawn around the edges of the hand that is found out by edge detection in the input image.we also fill the image by setting the thick- ness parameter of the contour to -1. Contouring helps us to identify the hand boundary which will help us to find the number of fingers which in turn will help in gesture identification.

Convex Hull

The convex hull of a set of points in the euclidean space is the smallest convex set that contains all the set of given points. The convex hull is similar to the shape formed by a rubber band stretched around a set of points. Convex hull is drawn around the contour of the hand, such that all contour points fall inside the convex hull. This makes an envelope around the hand contour.

Convexity Defects

A defect is present whenever the contour is far away from the convex hull. The convexity defects returns a vec- tor containing the start point and end point of the line of defect in convex hull and the point on the contour which is far away from the convex hull.

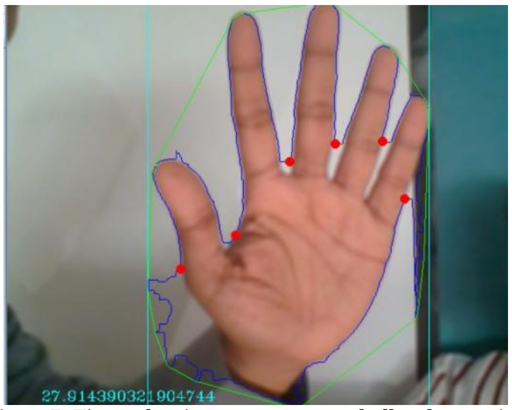


Figure 7: Figure showing contour, convex hull and convexity defects. Contour is shown in blue, convex hull in green and convexity defects are shown as red dots.

Gesture Identification

Back Propagation Algorithm

The Back Propagation algorithm is a supervised method for multi layer feed forward networks. The feed forward networks process the information and transfer output from one node to the next node, these nodes are called neurons. These neurons accept input from the den- drites that pass the electrical signal to the axon of the neu- ron. The axon passes the signal to the dendrite of the next neuron in the network through the synapse. The back propagation algorithm models a given function by modify- ing the weights of the input signal of a neuron to generate a desired output. The network is trained using a super- vised learning method where the error between the net- works output and the expected output is passed back to the previous layers in the network to update the internal weights based on the error values that is back propagation. The hidden layers in the network have n neurons in each layer , each with n+1 input weights 1 for each input column in data set. Neuron activation function is defined as weighted sum of the inputs i.e

Activation = sum(weight[i] * input[i]) + bias

Transfer function is generally used in sigmoid func- tion. It is used to transfer the activation value in the net- work. Sigmoid function is defined as

Output = 1/(1+e$^{-activation}$)

The sigmoid function graphically looks like an ‘S’ shape that is also called a logistic function.

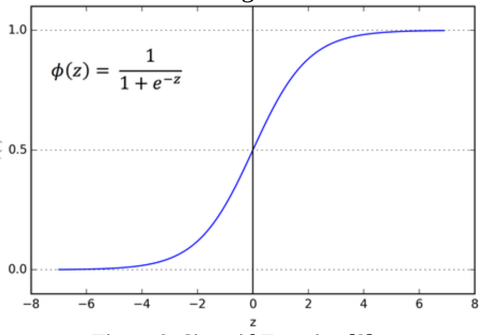


Figure 8: Sigmoid Function [8]

These function values are passed to the next layer of the network via forward_propogate function.
Transfer derivative function for each output, neuron its slope is calculated. We are using sigmoid function hence the derivative is calculated as
Derivative = output * (1.0- output)

Error back propagation The error is calculated at each output neuron i.e, this error signal is then propagated back- wards in the network and is given as:
error = (expected-output) * trans- fer_derivative(output)
Where expected is the expected value for the neu- ron, output is the output value for the neuron and trans- fer derivative() calculates the slope of the neurons output value. The error calculation is used for neurons in the out- put layer. The expected value is the class value itself. The error signal in the hidden layers are calculated as weighted errors of each neuron in output layers.
error = (weight[k] * error[j]) * trans- fer_derivative(output)

Convolutional Neural Network
Our project predicts the hand gestures using 2D Con- volution Neural Network. This Network runs over Ten- sorflow background using Keras as the major supporting library for Training and Prediction of the Neural Network. A unique feature of our Prediction model is that when- ever a new Gesture needs to be added, we can add it with- out the need to import a new Gesture Dataset for Training the Network, as we can generate Dataset and use it for
Training the neural network.
After the Image Processing is done on the images captured, like edge detection, hand contour analysis, etc. Those Images are then passed to the neural network. Note that the size of each image that is passed to the Neural Network is '100*100' pixels,
This is done to reduce the time complexity of the Training and Prediction steps. The input to Neural Net- work is provided is a 100*100 pixel RGB image. This implementation makes use of 4 layers i.e 1 input layer, 2 hidden layers and 1 output layer.
Training the neural network Training process includes training the CNN for gesture class prediction. Training is performed on a single GeForce 940MX graphics card. The main limit here is the device's memory capacity as described training and testing datasets exceed a memory capability. The proposed model is evaluated on the dataset consisting of 15,000 data samples obtained from 3 differ- ent 3D-models of the human hand.
For each model 10 different hand gestures have been created. Each of the gestures is represented by 1500 snap- shots summing up to 15,000 data samples per 3D-model. This would result in the transformation and storage of 15000 data samples by the Tensorflow library during train- ing for 100-on-1 cross-validation (based on samples not on models), including weights as well as the subsequent im- age transformation steps, which is more than the device can store during the training phase.

IV. CONCLUSION

In our system, we have implemented various image processing techniques to fully extract the hand from the background and use it as input to the deep neural network for predicting the gestures. These specified techniques can be used accurately and efficiently by applying some con- straints.

ACKNOWLEDGMENTS

We would like to thank our HOD Dr. Satish Y. Ket and our Computer Engineering Department for their support and guidance during the implementation of this project and also while writing this paper.
[1, 7, 9–15]

REFERENCES

[1] "OpenCV usage documentationhttp://docs.opencv.o rg."
[2] "Python." [Online]. Available: https://en.wikipedia. org/wiki/Python_(programming_language)
[3] [Online]. Available: https://en.wikipedia.org/wiki/ Python_Imaging_Library
[4] "Numpy." [Online]. Available: https://en.wikipedia. org/wiki/NumPy
[5] [Online]. Available: https://en.wikipedia.org/wiki/ TensorFlow
[6] "keras." [Online]. Available: https://en.wikipedia. org/wiki/Keras
[7] "Background subtraction using opencv." [Online]. Available: http://opencvpython.blogspot.com/2012/ 07/background-extraction-using-running.html
[8] "Activation Functions in Neural Networks." [Online]. Available: https://towardsdatascience.com/activation-functions- neural-networks-1cbd9f8d91d6
[9] S. K. Yewale and P. K. Bharne, "Hand Gesture Recogintion using Different Algorithms Based on Artificial Neural Network," IEEE, 2011.
[10] G. R. S. Murthy and R. S. Jadon, "Hand Gesture Recognition using Neural Networks."
[11] S. Hussain, R. Saxena, X. Han, J. A. Khan, and P. H. Chin, "Hand Gesture Recognition Using Deep Learning."
[12] Dhawan and V. Honrao, "Implementation of Hand Detection Based Techniques for Human Computer Interaction."
[13] Z. Yang, Y. li, W. Chen, and Y. Zheng, "Dynamic Hand Gesture Recognition Using Hidden Markov Model," in 7th International Conference on Com- puter Science and Education(ICCE 2012).
[14] "Back Propagation Algorithm." [Online]. Available: https://www.analyticsindiamag.com/6-types-of- artificial-neural-networks-currently-being-used-in- todays-technology/
[15] "Contours, Convex Hull and Convexity Defects." [Online]. Available: http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html

★★★