*Turkish Journal of Computer and Mathematics Education Vol.12 No.11 (2021), 5609-5616*
*Research Article*

# Hand Gesture Recognition System using Convolutional Neural Network

**Devendra Singh[a], Gaurav Das[b], Saad Y. Sait[c]**

[a]Department of Computer Science and Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, TamilNadu, India. E-mail: devu2456@gmail.com
[b]Department of Computer Science and Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, TamilNadu, India. E-mail: gauravgdas@gmail.com
[c]Department of Computer Science and Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, TamilNadu, India. E-mail: saady@srmist.edu.in

**Article History**: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

**Abstract:** Gestures provide the ability to interact with machines efficiently through Human computer interaction; this can be accomplished by creating interfaces which perform gesture recognition, thereby automating tasks. Hand gesture recognition is a type of gesture recognition that is very useful to automate tasks for the specially challenged. Challenges in this area are complex background, camera angle, and illumination. It uses GMM technique for background subtraction and VGG16 architecture, trained on images captured using a camera, to achieve a fast and robust classification system for gesture recognition, as demonstrated by experiments. An F1-score above 92% is obtained for all classes.

**Keywords:** Convolutional Neural Network (CNN), Human Computer Interface (HCI), Hidden Markov Model (HMM), Pseudo Two-dimension Hidden Markov Model (P2- DHMMs), Gaussian Mixture Model (GMM).

## 1. Introduction

Hand gestures are powerful and effective communication mode for Human Computer Interaction (HCI). Mechanical devices are used with computers like mouse, keyboard, joystick but these do not provide a natural interface for the interaction between device and user. Our system therefore consists of webcam and laptop interface for capturing of hand gesture and going through segmentation, augmentation and finally classification to recognize a gesture. Sensor based gesture recognition is done by using data-glove and other sensory devices whereas the vision-based method only requires a camera. More specifically, we believe the latter to be useful for the specially challenged. The challenging problems of the system are complex backgrounds, variation in lighting, robustness to variation in person and camera angle in order to achieve real time performance. Segmentation is finding patterns and connection within pixels with a specific property such as color and intensity.

In the modern world where the technology is growing so rapidly, hand recognition technology can be really useful as day to day the technology is improving giving us really less time to think what is old and what is new. In the future hand recognition technology will be very useful in controlling our day-to-day activities.

The basic goal of the technology is to recognize the human gestures using machine learning algorithms. The camera quality also plays a big role in the process of recognizing the captured gesture. Due to this small innovation which can be used in a mobile phone to many other devices and can be really useful in the coming days.

## 2. Literature Survey

There are various techniques and algorithms used for gesture recognition and few relevant are described as follows. [1] proposes skin detection, expansion and corrosion and then all contours are extracted (i.e., face, hand) and contour of hand is found by using VGGNet network and finally using pyramidal pooling module gesture is recognized. This pooling method is used with attention mechanism to increase the receptive field and classify more efficiently. The input passes through convolutional layer of 3x3, then four different spatial pyramids are pooled to obtain the size of 1/4,1/8,1/16,1/32 feature map. A technique to recognize hand gesture in sequence videos is proposed in [2] to find an architecture of CNN and Long short term memory to achieve dynamic recognition of gestures with high accuracy. The frames are chosen from the video and Convolutional network is used to extract features and then these frames are labelled and classified. The RNN-LSTM is used for classification of gesture by their sequential parameter of frames. In [3] execution is done by applying Gray world's Algorithm for illumination compensation and then Converting the image from RGB to YCbCr to detect Skin and marking Skin Pixels with Blue and Image Segmentation to reduce the computational time needed for the processing of the image. In Image Filtering technique, the value of image is found by applying algorithm to the value of the adjacent pixels. For better detection normalization is done at every step also called as cross

correlation algorithm. This method is simple and less processing is required and works well with static gestures recognition.[4] proposes PCA algorithm and feature extraction is done to create a gesture training database so that all the collected data could be kept there and later stored so that the algorithm can use it to test whether the gesture should work on the given circumstances. Classification of data is also performed where each and every data is classified based on the data received and it's matched with the data in database to give accurate result. The algorithm shows that when gesture is entered which is already stored in the database the projection distance is minimal thus excellent but for unknown gesture the system does not respond. The segmentation technique HS-ab and PCA are used effectively and shows PCA is sensible for the use of dynamic gesture recognition.[5] In this paper the proposed method is using HMMs. HMM is represented by vectors and matrix's where gesture recognition needed appropriate HMM parameters and the selection process are called HMM training. New gestures can be easily added by only re-training another HMM as the relationship between new model and original model is independent. [6] this paper uses the P2-DHMMs.This approach is used for the first time in hand gesture recognition framework. For each gesture there is a pseudo hidden Makarov model. To determine probability of each model Viterbi algorithm is used. Compared to a template-based method this system offers more flexible framework for gesture recognition and offers promising results in difficult recognition.

## 3. Proposed Work

The proposed Gesture Recognition System provides remote access to electronic device by detecting the gesture given as input by the user. Objective is to build an easy and comfortable method which can provide remote access to device without any physical interaction. The System uses Gaussian mixture model for background reduction and Convolution Neural Network (CNN) and the architecture used is VGG16.The data consist of various images of hand gestures at different background, illumination and angle. The data is collected using a camera. Only the hand gesture features are collected by Background Subtraction using GMM. The collected images are then pre-processed and fed to the neural network and classified according to the features and each class is interpreted with an action, here it is volume control, next, previous, close and other basic commands. The structure of the system is shown in Fig.1.

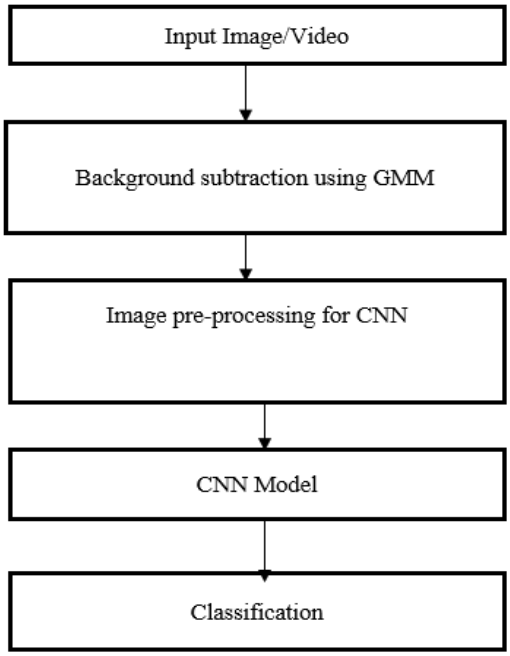


**Figure 1.** Structural Diagram of Gesture Recognition System

## 4. Background

CNN is an artificial neural network which has the ability to recognize pattern and is mostly applied in image recognition problems. Convolutional neural networks are different from the regular neural network as in regular neural network the hidden layer nodes are fully connected with the previous layer whereas in CNN they are connected to nearby nodes and each node has same weight. Convolution is used for simplifying a complex image data so it is easier to understand and analyze. CNN consist of convolutional layers and the pooling layers with fully connected neural network. The activation function used is Relu which keeps the dimensionality of the data. Then the data is flattened and passed through the fully connected layers with SoftMax function which classifies

the data accordingly. To understand CNN, we have to understand every layer used in the network and all the building blocks such as kernel, stride, padding, pooling and flatten, their importance and the functionality and how they reduce the dimensionality of any input while keeping all the important features. We can divide CNN into two parts- Feature Learning and classification. Feature learning consist of convolutional layers and pooling layers and classification consist of flatten and fully connected layer with SoftMax function for multiple classes classification.

### 4.1. The Convolutional layer

It is the main component of a Convolutional neural network. It basically works by using a filter also called as kernel over the pixels. Kernel is used to extract the features and reduce dimensionality by applying the filter over input data iteratively and performing dot product in the sub-region and produce a matrix of dot products of the pixel's values of the data. The basic example is given in the fig 2.

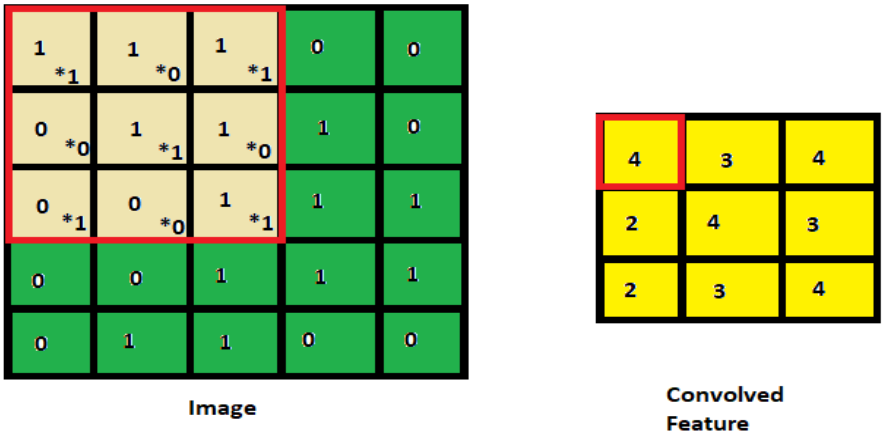


**Figure 2.** Feature extraction using Kernel

We have taken a small size (5*5) matrix for easy explanation of the functionality of kernel with kernel size (3*3), here kernel/filter= [{1,0,1}, {0,1,0}, {1,0,1}}. The kernel weight is determined by the CNN on itself and here we iterate 9 times to completely cover the image. This is a default stride value of 1. Padding can also be used which extends the size of input matrix by placing rows and columns around the original matrix with value 0. The size of the output matrix depends on the value of input size, kernel size, stride and padding by using a formula-

$$O = \left[ \frac{I-K+2P}{S} \right] +1$$

Here O & I represent Output & Input size of the feature matrix with K being the kernel size and P & S represents the Padding and Stride values.

In our example I=5, K=3, P=0, S=1.Thus the output feature map size is 3.

$$O = \left[ \frac{5-3+2*0}{1} \right] +1 = 3$$

This process results in a convolution feature map which detects features and with use of multiple convolutional layer it can determine more specific features like edges, shapes, objects, etc. which otherwise may not be found.

The convolutional layers are used with ReLu activation function as it increases the non-linearity in the image data. This function rectifies the data by eliminating all the negative pixel values replacing them with 0 and the output contains only non-negative values.

### 4.2. The Pooling layers

It is used to reduce the amount of variables in the feature map by reducing its size enabling faster processing by two ways either by max-pooling which takes the maximum value in a feature matrix or average pooling which takes the average value of the feature matrix in the pooling kernel. This is done to reduce the dimensionality but preserving all the important features of the input image. This is an example of max-pooling given in Fig.3

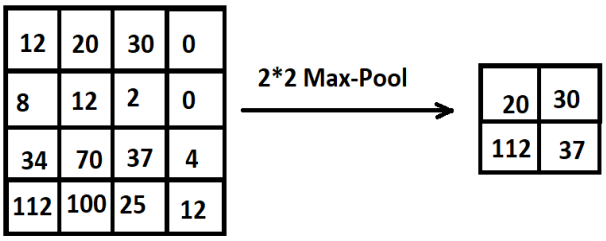


**Figure 3.** Max-pooling Downsizing

This combination of convolutional and pooling with the ReLu activation completes our Feature Learning process and now we move towards classification. Now to classify the data we need fully connected layers thus requiring to flatten the feature matrix. Flattening transforms the pooled feature matrix into a single dimensional data for feeding it to the dense layers. The dense layer then classifies the data. The complete CNN architecture is show in fig.4.

There are various ways to improve the efficiency and accuracy of the CNN architecture by modifying the number of kernels, convolutional layers, pooling layers. Some pre-defined architectures are used as they seem to be more accurate and robust than other i.e., AlexNet, VGGNet etc.

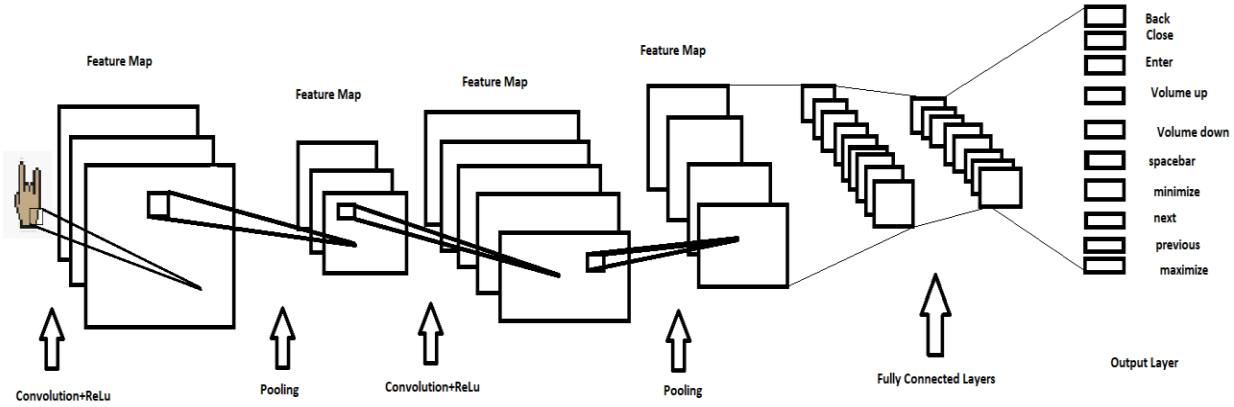


**Figure 4.** Complete CNN Architecture

We have used The VGGNet 16 architecture which is designed by Simonyan and Zisserman and is highly preferable and easy to use. It consists of 3 pairs of convolutional layers with filter 32,64,128 filters of size 3*3 and each pair having a max-pooling layer of size 2*2 in addition with Dropout layer which randomly drops nodes in the output. The hyperparameters are fixed with activation function relu which then flattened to be fed to dense layer of 128 nodes connected to final output layer with SoftMax regression classifying into 10 classes.

## 5. Implementation

The Gesture Recognition System consist of four modules which are implemented in the order Data Collection, Image Pre-processing, Training & Testing, Gesture Recognition.

### 5.1. Data Collection

We take input of images from the webcam and using GMM algorithm we subtract the background. The Gaussian Mixture Model uses reference image as background and anything moving is considered as a foreground. These images are saved in 10 different folders for each gesture.

### 5.2. Image pre-processing

We extract the hand skin and crop around the counterpart, then converted to threshold image and resized into 100*100 pixels.

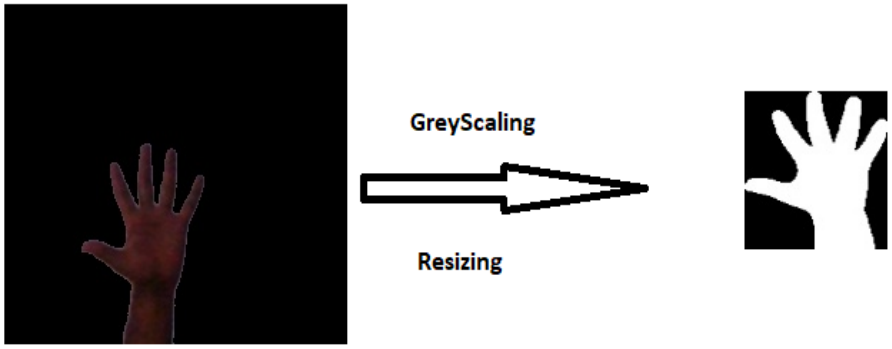


**Figure 5.** Image pre-processing

### 5.3. Training & Testing

In this module we pass the pre-processed images into a VGG16 architecture containing group of Convolutional layers, Max-Pooling layers and Dropouts which are then flatten and passed through fully connected neural network. The training dataset consist of 1000 images for each gesture and 200 each for validation and test set thus totalling 10000 images for training and 2000 for validation and testing. The number of epochs is 20 and steps per epoch are considered as total images upon batch size. The model is trained and charts are plotted. VGGNet takes more time compared to simple CNN model but it is more accurate and preferred by the developers. The optimizer used is Stochastic Gradient Descent with learning rate 0.1 to improve the accuracy as it uses random dataset for each iteration rather than the whole dataset.

This model architecture contains 100*100 size data fed through the pair of convolution layer (kernel size 32) and one max pooling layer to reduce the size of features by half. This is repeated with CNN layers of kernel 64,128 then to be flattened to be fed to fully connected dense layers. The final result is evaluated using fully connected network and SoftMax activation function. This structure can handle high resolution data and cost effectively increasing the accuracy. Fig.6 shows the layer architecture used in the implementation.


**Figure 6.** VGGNet16 Detailed summary

### 5.4. Gesture Recognition

In this module the trained model is loaded and used on the real-time images that are captured from the webcam and classifies the gestures into 10 different classes which are associated with a specific action. The gestures are associated to press different key combination to command the laptop. For now, the basic actions are back, enter, close, maximize, minimize, volume up, volume down, next, previous, close. The action is performed after a pause of 5 seconds to enable users to get enough time to change the hand gesture accordingly. The application is easy to use and performs the same data pre-processing before using the loaded model weights and can be reused with any number of class objects to provide key combinations that can be useful for any sort of purpose.

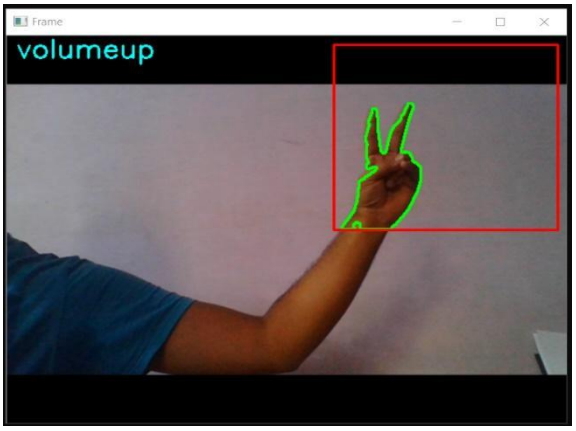


**Figure 7.** Application Interface

To validate the model, we created an application to interact with the laptop system using 10 different gestures as shown in diagrams: -

**Figure 8.** Gestures classes

The application loads the model and uses it to recognize real time gestures provided by the user to interact with system. The system is robust with minimal recognition time thus able to provide a natural interaction with ease and comfort.

In our experiment the 640*240-pixel webcam is used for input on a 2.6ghz processor with 8GB ram.

## 6. Results Discussion

In the proposed method 2 different types of architectures are used for Convolutional neural network for training the model. First One is a simple CNN Max-pooling architecture with validation accuracy of 93%. Fig.9 shows the accuracy & loss charts.

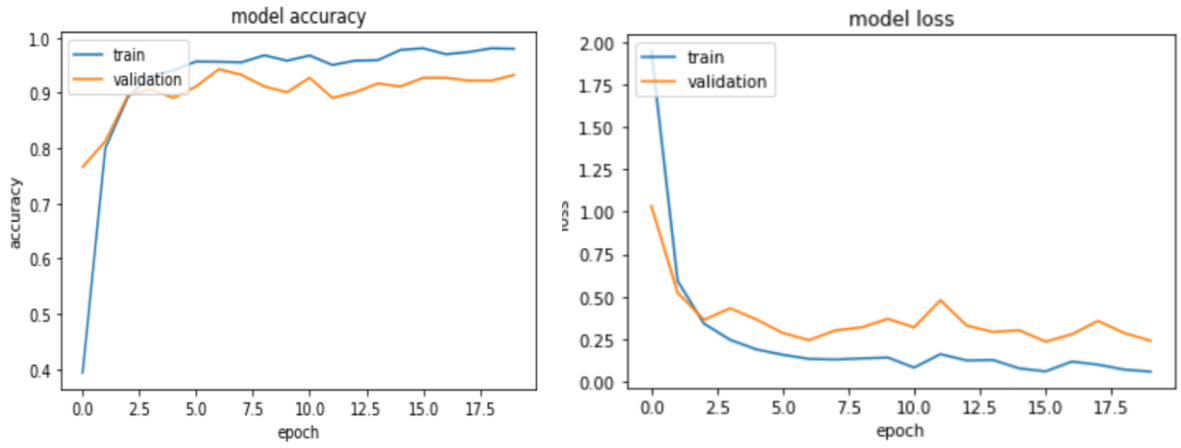


**Figure 9.** CNN Max-pooling architecture accuracy & loss

The second is a VGG16 architecture and its structure was given in Figure 6. With this sequential model we were able to improve the accuracy to 98%.

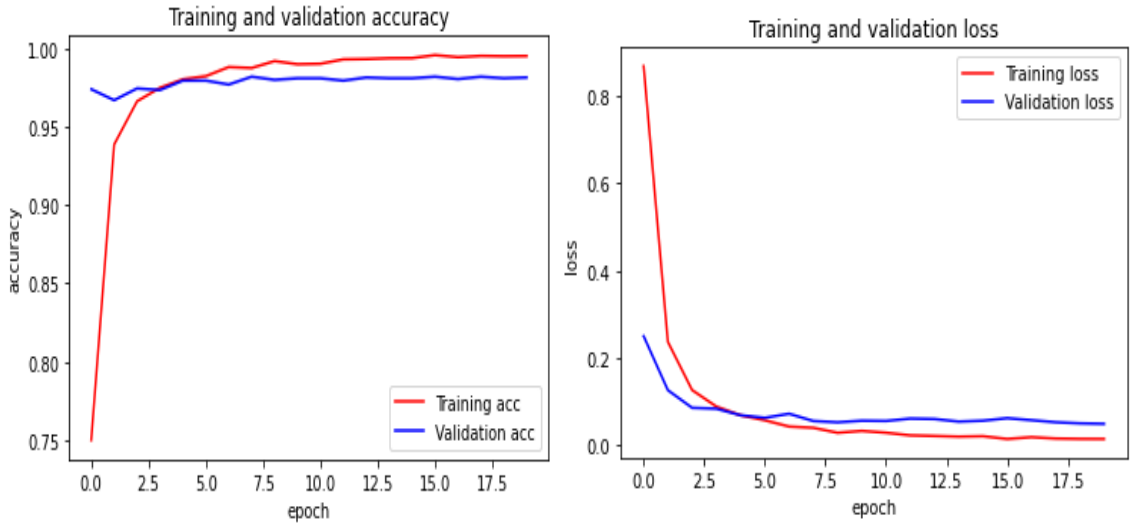


**Figure 10.** VGG16 architecture accuracy & loss

The evaluation on test data is done to verify the accuracy which came out to be 98.2%. The charts show in fig.10 and test data evaluation suggest that model is not overfitted and the precision and recall data is shown in Table 1 and confusion matrix is shown in Figure 11.

**Table 1.** Precision and Recall

| Classes | Precision | Recall | Fi-score |
|---|---|---|---|
| Back | 1.00 | 0.98 | 0.99 |
| Close | 1.00 | 1.00 | 1.00 |
| Enter | 1.00 | 0.91 | 0.95 |
| Maximize | 0.98 | 1.00 | 0.99 |
| Minimize | 0.94 | 1.00 | 0.97 |
| Next | 0.96 | 0.99 | 0.98 |
| Previous | 1.00 | 1.00 | 1.00 |
| Spacebar | 1.00 | 1.00 | 1.00 |
| Volume down | 0.97 | 0.95 | 0.96 |
| Volume up | 1.00 | 1.00 | 1.00 |

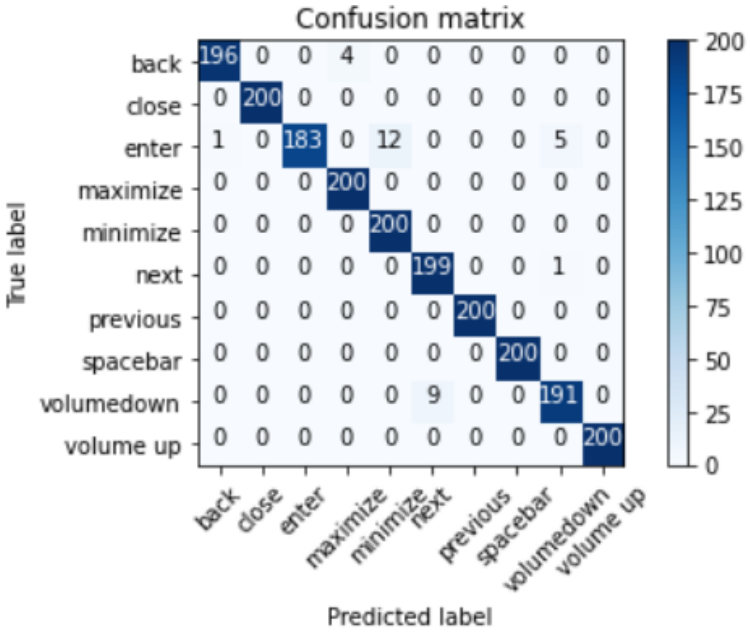


**Figure 11.** Normalized Confusion Matrix

The Confusion Matrix provides us with useful data of which gestures are easily recognized and which gestures produce similarities reducing the accuracy of the model. This helps in collecting good data and the saturation limit of number of classes. Here as we have used 10 classes and enter seem harder to detect showing similarity to minimize and volume down. Also, the back and maximize have a little similarity as shown by the figure 11. These can be solved by changing the gesture so that less similarity occurs in training. Other gestures are being easily classified.

## 7. Conclusion

Our proposed model combines GMM technique for background subtraction and VGG16 architecture to achieve a real-time robust system that can recognize the gesture from a varying distance for different people, background and intensities of light with accuracy and ease. Results are reliable and it can work with any computer-controlled devices having an input camera. Specifically, we believe that it would make it easier for the specially challenged to interact with computer-based systems using gestures. The confusion matrix suggests that the model can easily handle a fair number of classes without sacrificing the accuracy.

## References


1. Huang, H., Chong, Y., Nie, C., & Pan, S. (2019, June). Hand gesture recognition with skin detection and deep learning method. *In Journal of Physics: Conference Series*, (Vol. 1213, No. 2, p. 022001). IOP Publishing.
2. Kadam, S., Ghodke, A., & Sadhukhan, S. (2019, April). Hand Gesture Recognition Software Based on Indian Sign Language. In 2019 1st *International Conference on Innovations in Information and Communication Technology (ICIICT)*, (pp. 1-6). IEEE.
3. Contreras Alejo, D.A., & Gallegos Funes, F.J. (2019). Recognition of a Single Dynamic Gesture with the Segmentation Technique HS-ab and Principle Components Analysis (PCA). *Entropy*, *21*(11), 1114.
4. Chen, F.S., Fu, C.M., & Huang, C.L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and vision computing*, *21*(8), 745-758.
5. Binh, N.D., Shuichi, E., & Ejima, T. (2005). Real-time hand tracking and gesture recognition system. *Proc. GVIP*, 19-21.
6. He, Y., Yang, J., Shao, Z., & Li, Y. (2017, July). Salient feature point selection for real time RGB-D hand gesture recognition. In 2017 IEEE *International Conference on Real-time Computing and Robotics (RCAR)*, (pp. 103-108). IEEE.
7. Elsayed, R.A., Sayed, M.S., & Abdalla, M.I. (2017, December). Hand gesture recognition based on dimensionality reduction of histogram of oriented gradients. In 2017 *Japan-Africa Conference on Electronics, Communications and Computers (JAC-ECC)*, (pp. 119-122). IEEE.
8. Saha, H.N., Tapadar, S., Ray, S., Chatterjee, S.K., & Saha, S. (2018, January). A machine learning based approach for hand gesture recognition using distinctive feature extraction. In 2018 *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 91-98). IEEE.
9. Patel, U., & Ambekar, A.G. (2017, August). Moment Based Sign Language Recognition for Indian Languages. In 2017 *International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, (pp. 1-6). IEEE.