ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 6 June, 2014 Page No. 6395-6401

Implementation of Neural Network based Hand Gesture Recognition System

***Neha V. Tavari*[1], *Prof. A. V. Deorankar*[2]**

[1]Department of Computer Science and Engineering,
Government College of Engineering, Amravati, India
tavarineha@gmail.com

[2]Department of Computer Science and Engineering,
Government College of Engineering, Amravati, India
avdeorankar@gmail.com

Abstract: ***Communication is the means of exchanging information, views and expressions among different persons, in both verbal and non-verbal manner. Hand gestures are the non verbal method of communication used along with verbal communication. A more organized form of hand gesture communication is the sign language, which is the language of communication for deaf and dumb people. Most of these physically impaired communities are dependent on sign language translators to express their thoughts to rest of the world. This causes isolation of these people in society. Hence, Sign Language Recognition (SLR) is one of the most growing fields of research today. Hand gesture is an active area of research in the computer vision, mainly for the purpose of sign language recognition and Human Computer interaction. Here, I have developed a system for hand gesture recognition of Indian sign language, in which each alphabet and numbers of the english vocabulary is assigned a sign which is used by deaf and the dumb to communicate with each other. An artificial neural network based classification with error backpropagation algorithm is used for recognizing different signs and translate them into text and voice format. To implement this approach we have utilized a simple web camera to capture hand gesture images. Thus presenting a system that recognizes Indian sign language (ISL) based on hand gestures allows the user to interact with the system in natural way.***

Keywords: Computer Vision, Deaf and dumb, Human computer interaction, Indian Sign language.

1. Introduction

Sign language is widely used by physically impaired people who cannot speak and hear or who can hear but cannot speak and is the only medium of communication for those people. It uses gesture instead of sound to convey meaning. It is nothing but the combination of hand shapes, movements and orientations of hands or body, facial expressions and lip-patterns for conveying messages. These gestures are widely used by the deaf and dumb people to express their thoughts [1]. Usually physically impaired people needs the help of sign language interpreters for translating their thoughts to normal people and vice versa . But it becomes very difficult to find a well experienced and educated translator for the sign language every time and everywhere in daily life, but human-computer interaction system for this can be installed anywhere possible. So a system recognizing the sign language gestures (HCI) automatically is necessary which will help to minimize the gap between deaf people and normal people in the society. This resulted in the development of automatic sign language recognition systems which could automatically translate the signs into corresponding text or voice without the help of sign language interpreters. Moreover hearing people have difficulties in learning sign language and likewise the majority of those people who were born deaf or who became deaf early in life, have only a limited vocabulary of accordant spoken language of the community in which they live. Hence a system of translating sign language to spoken language would be of great help for deaf as well as for hearing people.

As sign language is well structured code gesture, each gesture has a meaning assigned to it [2]. There are number of sign languages spreaded across the world. The sign language used by those deaf and mute at a particular place is dependent on the culture and spoken language at that place. American Sign Language (ASL), British Sign Language (BSL), Japanese Sign Language family (Japanese, Taiwanese and Korean Sign Languages), French Sign Language family (French, Italian, Irish, Russian and Dutch Sign Languages), Australian Sign Language, etc. [3] are the examples of regionally different sign languages. Most of the researchers in this area concentrate on the recognition of American Sign Language (ASL) since most of the signs in ASL are single handed and thus, complexity is less. Another attractive feature is that ASL already has a standard database that is available for use. On the other hand, Indian sign language (ISL) is used by the deaf and dumb community in India. It consists of both word level gestures and fingerspelling which is used to form words with letter by letter coding. The words for which no signs exist can be expressed with the use of letter by letter signing. It helps in recognizing the words for which the signer does not know the gestures or to emphasis or clarify a particular word. So the fingerspelling has key importance in sign language recognition. Since ISL got standardized only recently and also since tutorials on ISL gestures were not available until recently, there are very few research work that has happened in ISL recognition [4]. Here we propose a method for hand gesture recognition of Indian sign language alphabet and numerals.

2. Proposed System

The proposed work is aimed to develop a sign language education and recognition platform for hearing impaired peoples and communication system for dumb people to convey their message. The main approaches for analyzing and classifying hand gestures for HCI include Glove based techniques and Vision based techniques. The objective my work is to build a system that uses natural hand gestures as a modality for recognition in the vision-based setup. Our approach has following operational flow and it is illustrated in figure 1.

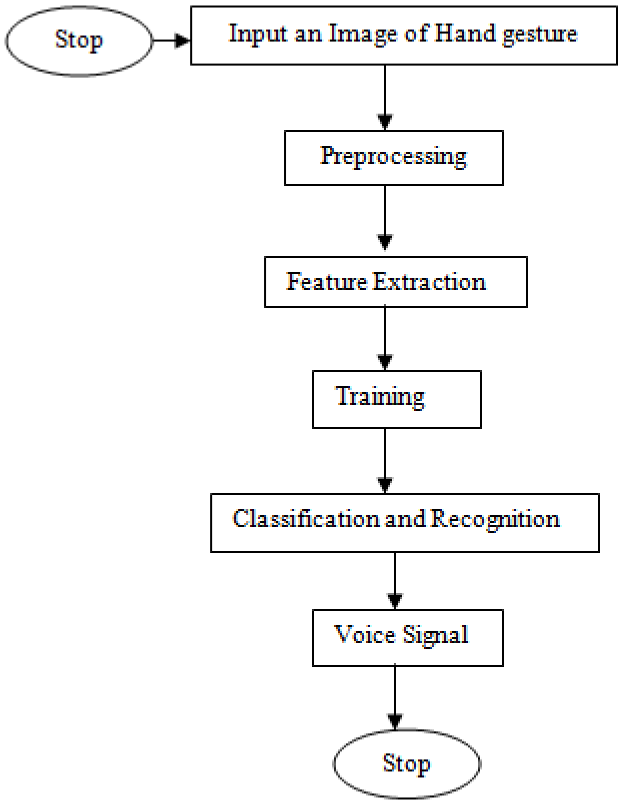


Figure 1: Flow of Hand Gesture recognition system

2.1 Database description

Image acquisition is the process to capture the hand gesture images which represents different signs. The resolution of various image capturing devices may not be the same. This results in different resolution of the captured images. For accurate comparison of the features and to reduce the computational effort needed for processing, all the images should be scaled to a uniform size [1]. The image database is created for training and testing phase. As the image dataset of Indian sign language alphabet and numerals are not available from any resources, it is made available with suitable lighting and environmental setup. The images for training and testing are captured in a white background with webcamera. The database consist of 36 hand gesture of ISL i.e. 26 English alphabet and 9 numbers. The system works offline recognition i.e. we give test image as input to the system and system tells us which sign is recognized. Indian sign language alphabet and numerals are shown in figure 2 and figure 3 respectively.

Figure 2: Representation of ISL alphabets

Figure 3: Representation of ISL numerals

`

2.2 Preprocessing

The image scene and information should not be altered by local changes due to noise and digitization error. Hence to satisfy the environmental scene conditions, preprocessing of the raw data is highly important. Preprocessing consist of two steps:

- Segmentation
- Morphological filtering

The objective of gesture segmentation is to extract the gesture region from the background of the image. Hand segmentation is the process of extracting the hand sign from the captured image and also gesture region is extracted from the background of the image. The segmentation process depends on the type of gesture, if it is dynamic gesture then the hand gesture need to be located and tracked , if it is static gesture the input image have to be segmented only [5].

The result of segmentation produces a binary image with the skin pixels in white color and background in black color. The resulting binary image may contain noise and segmentation errors. Filtering and morphological operations are performed on the input image to decrease noise and segmentation errors if any. Image preprocessing includes the set of operations on images whose goal is the improvement of the image data that suppresses undesired distortions or enhances some image features important for further processing.

Figure 4: Removal of Background

Segmentation is performed to convert gray scale image into binary image so that we can have only two object in image one is hand and other is background. Otsu algorithm [6] is used for segmentation purpose and gray scale images are converted into binary image consisting hand or background.

2.3 Feature Extraction

Good segmentation process leads to perfect features extraction process and the later play an important role in a successful recognition process [5]. There are many interesting points on every object which can be extracted to provide a "feature" description of the object. Features vector of the segmented image can be extracted in different ways according to particular application. Under different scene conditions, the performance of different feature detectors will be significantly different. The nature of the background, existence of other objects (occlusion), and illumination must be considered to determine what kind of features can be efficiently and reliably detected. For the recognition of ISL, an algorithm to find Histogram of Oriented Gradient is implemented.

Histograms of Oriented Gradients:

Histogram of Oriented Gradients (HOG) are feature descriptors used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. HOG features are calculated by taking orientation histograms of edge intensity in local region [7].

A. Gradient Computation:

The first step of calculation is gradient values computation. The most common method is to simply apply the 1-D centered, point discrete derivative mask in one or both of the horizontal and vertical directions. Specifically this method requires filtering the gray scale image with the following filter kernels.

For the x direction, X = [0 -1 1]

For the y direction, Y = [0 1 -1]

Then convolution is performed on image vectors to form gradient in x and y direction

$$Dx = I * X \text{ and } Dy = I * Y$$

Let us consider hand gesture image for alphabet 'A' and apply gaussian filtering on this image to get filtered image as shown in figure 5.

fim = imfilter (img, myfilter, 'replicate', 'conv')

Figure 5: Filtered Image for Hand Gesture 'A'

The convolution operation is performed on binary format of an image i.e Figure 6 with the help of derivative filters in horizontal and vertical directions to find gradient in x and y direction respectively.

Figure 6: Gesture Image in Binary Format

Dx = convn (fimbin, x, 'same');

Dy = convn (fimbin, y, 'same');

The gradient images Dx and Dy are shown in figure 7.

Dividing the two resulting gradient vectors Dx and Dy element by element and performing atan ($\tan^{-1}$) will result in gradient orientation. Thus, magnitude and orientation of gradient is given as follows:

$$|G| = \sqrt{Dx^2 + Dy^2}$$

$$\Theta = arctan\ (Dy/Dx)$$

Figure 7: Gradient of Image in x and y Directions

The function atan () returns a value in the interval (-π, π). So the orientation of the gradient, for a pixel is Θ radians. The orientation in radian is converted to degrees

$$\alpha = \Theta * 180 / \pi$$

B. Orientation Binning:

The second step of calculation involves creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation based histogram channel based on the values found in the gradient computation. Angle histogram is created which is a polar plot showing the distribution of values grouped according to their numeric range. Each group is shown as one bin.

In continue with the above example, after finding convolution, we will get gradient representation of image 'A'. Now applying atan() on gradient, we get,

Theta=atan(gradient);

After converting values of radians into degrees and applying rose() function, it gives following orientation histogram.

rose(theta)

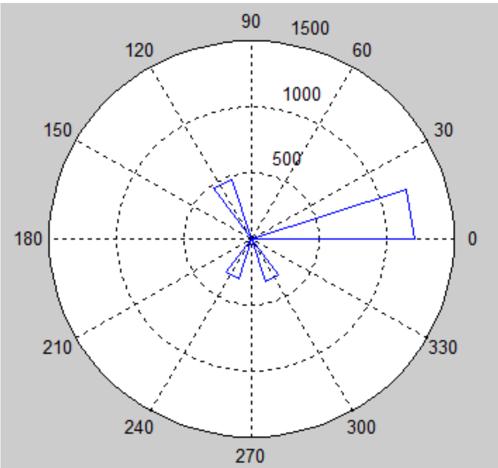


Figure 8: Orientation Histogram

2.4 Neural Network

Neural networks are composed of simple elements operating in parallel. These are inspired by biological nervous system. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

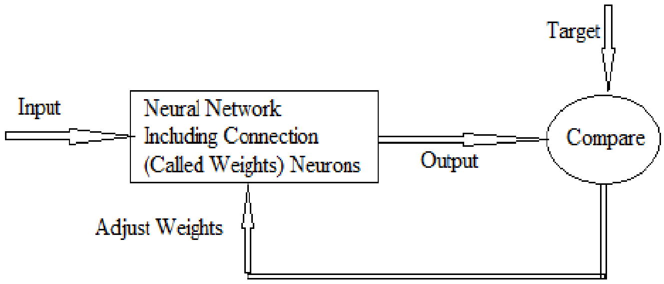


Figure 9: Neural Network Block Diagram

Many algorithms exist for determining the network parameters. In neural network literature the algorithms are called learning or teaching algorithms, in system identification they belong to parameter estimation algorithms. The most well-known are back-propagation and Levenberg-Marquardt algorithms. It is a supervised learning method, and is a generalization of the delta rule [8]. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks.

Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks. From a desired output, the network learns from many inputs, similar to the way a child learns to identify a dog from examples of dogs. It is a supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks. The goal and motivation for developing the backpropagation algorithm is to find a way to train multi-layered neural networks such that it can learn the appropriate internal representations to allow it to learn any arbitrary mapping of input to output.

The backpropagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training patterns input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Subtract a ratio (percentage) of the gradient from the weight.

This ratio (percentage) influences the speed and quality of learning; it is called the *learning rate*. The greater the ratio, the faster the neuron trains; the lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction. Repeat phase 1 and 2 until the performance of the network is satisfactory.

Following are the steps to implement neural network for recognition. The MATLAB commands used in the procedure are newff, train and sim.

- The structure of the network is first defined. In the network, activation functions are chosen and the network parameters, weights and biases, are initialized.

- The parameters associated with the training algorithm like error goal, maximum number of epochs (iterations), etc, are defined.
- The training algorithm is called.
- After the neural network has been determined, the result is first tested by simulating the output of the neural network with the measured input data. This is compared with the measured outputs. Final validation must be carried out with independent data.

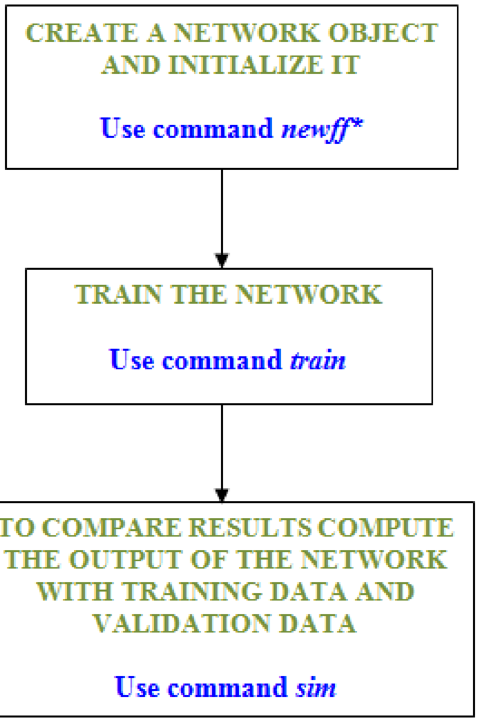


Figure 10: Basic Flow Diagram of NN Training

1) **Newff** - Create feed-forward back propagation network

The MATLAB command newff generates a MLPN neural network, which is called net.

net = newff (P, T, {S1 S2...SNl}, {TF1 TF2...TFNl}, BTF)

The inputs are,

P = R x Q1 matrix of Q1 sample R-element input vectors

T = SN x Q2 matrix of Q2 sample SN-element target vectors

Si = Size of ith layer, for N-1 layers, default = [ ].

TFi = Transfer function of ith layer. (Default = 'tansig' for hidden layers and 'purelin' for output layer.)

BTF = Backpropagation network training function (default = 'trainlm')

The default algorithm of command newff is Levenberg-Marquardt, trainlm. Default parameter values for the algorithms are assumed and are hidden from the user. They need not be adjusted in the first trials. Initial values of the parameters are automatically generated by the command.

2) **train** - Train neural network

After initializing the network, the network training is originated using train command. The resulting MLP network is called net1.

net1= train (net, P, T)

The arguments are,

net = the initial MLP network generated by newff

P = Network inputs

T = Network targets

3) **sim** - Simulate neural network

To test how well the resulting MLP *net*1 approximates the data, sim command applied. The output of the MLP network is simulated with sim command and called ytest.

ytest = sim (net1, x)

The arguments are

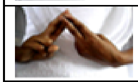
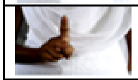
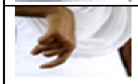
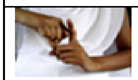
net1 = Final MLP

x = Input vector

3. Experimental Results and Discussions

The hand gesture dataset of Indian sign language consist of 216 images of 36 gesture, 6 samples each class with spatial resolution (640*480) pixel. The dataset is equally split into training and testing set. Training dataset is used to train the network. However testing dataset is used for testing the performance.

Table 1: Experimental result of the proposed System

Input image	Symbol in Database	No of test image	Recognized symbol
	'A'	3	'A'
	'B'	3	'B'
	'C'	3	'C'
	'D'	3	'D'
	'E'	3	'E'
	'F'	3	'F'
	'G'	3	'G'
	'H'	3	'H'
	'I'	3	'I'
	'J'	3	'J'
	'K'	3	'K'
	'L'	3	'L'

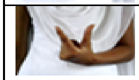
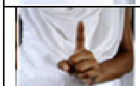
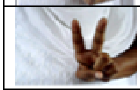
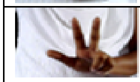
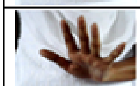
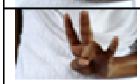
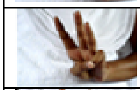
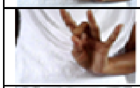
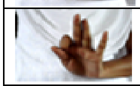
	'M'	3	'M'
	'N'	3	'N'
	'O'	3	'O'
	'P'	3	'P'
	'Q'	3	'Q'
	'R'	3	'R'
	'S'	3	'S'
	'T'	3	'T'
	'U'	3	'U'
	'V'	3	'V'
	'W'	3	'W'
	'X'	3	'3'
	'Y'	3	'D'
	'Z'	3	'B'
	'0'	3	'0'
	'1'	3	'1'
	'2'	3	'2'
	'3'	3	'3'
	'4'	3	'4'
	'5'	3	'5'
	'6'	3	'6'
	'7'	3	'7'
	'8'	3	'8'
	'9'	3	'9'

Table 2: Performance of the Proposed Method

Number of test images	Correctly classified	Wrongly classified	Accuracy(%)
36×3 = 108	99	9	91.66%

Accuracy = (Number of correctly classified signs/total signs) * 100
= (99/108)*100
= 91.66%

The performance of the gesture recognition algorithm is evaluated on the basis of its ability to correctly classify samples to their corresponding classes. From the table 1, it is seen that gestures x, y and z are misclassifed. Thus, the recognition rate can be defined as the ratio of the number of correctly classifies samples to the total number of samples.

From the above experiments, we can say that we have designed a system that was able to recognize different symbols of Indian Sign Language and we have removed difficulties faced by the previous works with improved recognition rate of 91.66%.

4. Conclusions

In today's digitized world, processing speeds have increased dramatically, with computers being advanced to the levels where they can assist humans in complex tasks. Yet, input technologies seem to cause a major bottleneck in performing some of the tasks, under-utilizing the available resources and restricting the expressiveness of application use. Hand Gesture recognition comes to rescue here. In this project, I have developed an Indian Sign Language (ISL) recognition system. The system is able to recognize 36 hand gestures of an Indian sign language which represents the alphabets from A to Z and numbers from 0 to 9. A sign language recognition system proposed for human computer interaction using Image processing technique is implemented successfully with accuracy comparable with those of recent application. It has been experimented with gesture images captured by a web camera and achieved satisfactory result with accuracy of 91.66%.

A neural network based method for automatically recognizing the hand gestures of Indian sign language is presented in this work. The feedforward backpropagation training algorithm is used to train the network. As the method is implemented completely by using digital image processing technique so the user does not have to wear any special hardware device to get features of hand shape. The proposed gesture recognition system can handle different types of hand gestures in a common vision based platform and it is suitable for both single handed and double handed gestures. Developing such system translating sign language to text/voice format will prove very useful for physically impaired people of india.

References

[1] Adithya V., Vinod P. R., Usha Gopalakrishnan ,"Artificial Neural Network Based Method for Indian Sign Language Recognition" , IEEE Conference on Information and Communication Technologies (ICT), pp. 1080-1085, 2013.

[2] Rajam, P. Subha and Dr. G. Balakrishnan , "Real Time Indian Sign Language Recognition System to aid Deaf

and Dumb people", 13th International Conference on Communication Technology(ICCT), pp. 737-742, 2011.
[3] Deepika Tewari, Sanjay Kumar Srivastava , “A Visual Recognition of Static Hand Gestures in Indian Sign Language based on Kohonen Self- Organizing Map Algorithm”, International Journal of Engineering and Advanced Technology (IJEAT), Vol. 2, Issue 2, pp. 165-170, Dec 2012.
[4] Manjusha U. C., “A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation”, International Journal on Computer Science and Engineering (IJCSE), Vol. 4, No. 03, pp. 406-415, March 2012.
[5] Rafiqul Zaman Khan and Noor Adnan Ibraheem , “Hand Gesture Recognition: A Literature Review”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 3, No. 4, pp. 162-174, July 2012.
[6] N.Otsu, “A Threshold Selection Method from Gray-Level Histograms”, IEEE transactions on systems, man, and cybernetics, Vol. 9, No. 1, pp. 62-66, January 1979.
[7] Takuya Kobayashi, Akinori Hidaka, and Takio Kurita, “Selection of Histograms of Oriented Gradients Features for Pedestrian Detection”, pp. 598–607, 2008.
[8] Jacek M. Zurada, “Introduction to Artificial Neural System”, pp. 185-190, 2006.

Author Profile

Neha V. Tavari, has received her B.E. degree in Computer Engineering from B.D. college of Engineering, Sewagram, India in 2011 and now she is pursuing M.Tech in CSE branch from Government college of Engineering, Amravati . Her area of research includes image processing, pattern recognition, and neural networks.

Prof. A. V. Deorankar, has received his M.E. degree in Electronics Engineering from Govt. College of Engineering, Amravati, India. He has published nine national level papers and seven international papers. He is also patent of one paper. His area of research includes Computer Network, Web Mining. Currently he is working as an Associate Professor at Govt. college of Engineering, Amravati, India.