International Journal of Computer Science and Network (IJCSN)
Volume 1, Issue 6, December 2012 www.ijcsn.org ISSN 2277-5420

Hand Gesture Recognition using Neural Network

[1]Rajesh Mapari, [2]Dr. Govind Kharat

[1] Dept of Electronics and Telecommunication Engineering, Anuradha Engineering College, Chikhli, Maharashtra-443201, India

[2] Principal, Sharadchandra Pawar College of Engineering, Otur, Maharashtra-443201, India

Abstract

This paper presents a simple method to recognize sign gestures of American Sign Language using features like number of Peaks and Valleys in an image with its position in an image. Sign language is mainly employed by deaf-mutes to communicate with each other through gestures and visions. We extract the skin part which represents the hand from an image using L*a*b* Color space. Every hand gesture is cropped from an image such that hand is placed in the center of image for ease of finding features. The system does require hand to be properly aligned to the camera and does not need any special color markers, glove or wearable sensors. The experimental results show that 100% recognition rate for testing and training data set.

Keywords: *Gesture recognition, boundary tracing, segmentation, peaks & valleys.*

1. Introduction

The ultimate aim of our research is to enable communication between speech impaired (i.e. deaf-dumb) people and common people who don’t understand sign language. This may work as translator [10] to convert sign language into text or spoken words. Our work has explored modified way of recognition of sign using peaks ad valleys with added feature of positioning of finger in image. There were many approaches to recognize sign using data gloves [11], [12] or colored gloves [15] worn by signer to derive features from gesture or posture.

Ravikiran J. et al. proposed a method of recognizing sign using number of fingers opened in a gesture representing an alphabet of the American Sign Language [1]. Iwan Njoto Sandjaja et al**.** proposed a modification in color-coded gloves which uses less color compared with other color-coded gloves in previous research to recognizes the Filipino Sign Language [2]. Jianjie Zhang et al. proposed a new complexion model has been proposed to extract hand regions under a variety of lighting conditions [3]. V.Radha et al. developed a threshold based segmentation process which helps to promote a better vision based sign language recognition system [4]. Ryszard S. Choras proposed a method identification of persons based on the shape of the hand and the second recognizing gestures and signs executed by hands using geometrical and Radon transform (RT) features [5]. Salma Begum, Md. Hasanuzzaman proposed system which uses PCA (Principal Component Analysis) based pattern matching method for recognition of sign [6]. Yang quan, Peng Jinye, Li Yulong proposed a novel vision-based SVMs [8] classifier for sign language recognition [7]. A vision based Sign Language recognition system uses many features of image like area, DCT and uses Neural Network [9] or HMM [14], [16].

2. Proposed Methodology

In this paper we present an efficient and accurate technique for sign detection. Our method has five phases of processing viz., image cropping, resizing, peaks and valleys detection, dividing image in sixteen parts, finding location of peaks and valleys as shown in Figure 1.

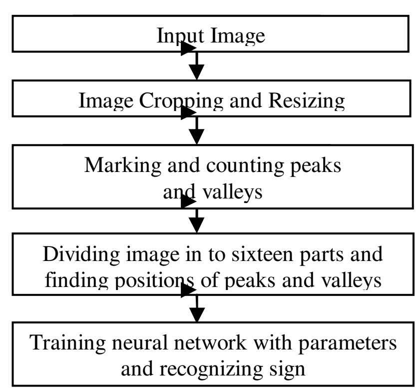

Fig. 1 Block Diagram of Sign Detection

Authors have collected data of 20 persons (students of engineering college) who have been given little training about how to perform signs. For acquiring image we have used camera of 1.3M pixels (Interpolated 12M pixels still image resolution).

In first phase we have read image and cropped it by maintaining height width ratio of hand portion only. Later on hand portion is resized to 256*256 size to extract features.

2.1 Cropping input image

First converts the RGB image to L*a*b* Color space to separate intensity information into a single plane of the image, and then calculates the local range in each layer. Second and third layer is intensity images are converted to black and white image according to threshold value of each layer. Two images are then multiplied to get one result image. From the result image 4-connected components are labeled. Properties of each labeled region are measured using Bonding box to make structures. Convert structure to cell array. Convert cell array of matrices to single matrix.

From this matrix hand portion is marked by marking square box on original RGB image.

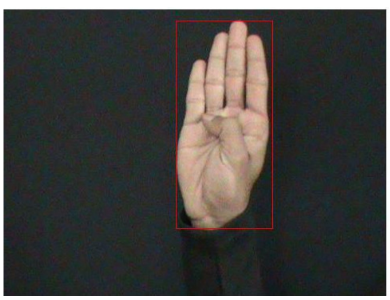

Fig. 2 Image of hand in with red box marked

If the width (W) hand portion is more than height (H) then cropping is W*W size else it is H*H size.

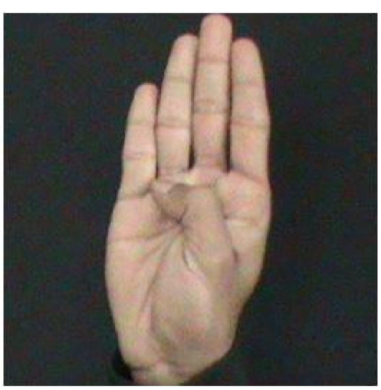

Fig. 3 Resized Image

2.2 Resizing image

After getting RGB image in size of either W* W or H*H , image is converted to gray scale image.
Image is then filtered using Gaussian filter with size [8 8] and sigma value 2 which found suitable for this experimentation.

Filtered image is then resized to 256*256 sized image. Hand portion image is then converted to 256*256 size RGB image, this way hand portion comes at the center of image. This way cropping operation is performed.

Fig. 4 Grayscale Image

2.3 Boundary Tracing for Peaks and Valleys

Resized image is smoothed by moving average filter to remove unnecessary discontinuities.

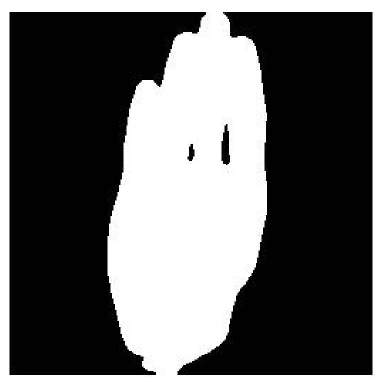

Fig. 5 Hand image before and after smoothing operation

Using morphological operations this smoothed image is converted to boundary image.

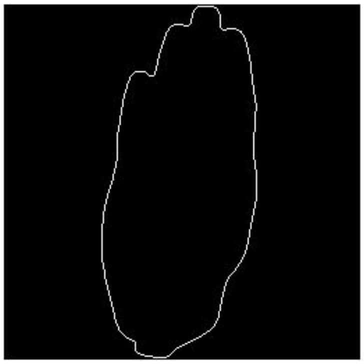

Fig. 6 Boundary Image

2.4 Peaks and valleys detection

After getting boundary image we first find the boundary tracing point from where to start and where to stop finding peaks and valleys. For this we find maximum value of x where white pixel exists.

We call this point as opti_x and then find corresponding value of y. The starting point on x direction as 0.80*x. From this x value we find starting y co-ordinate of starting point.

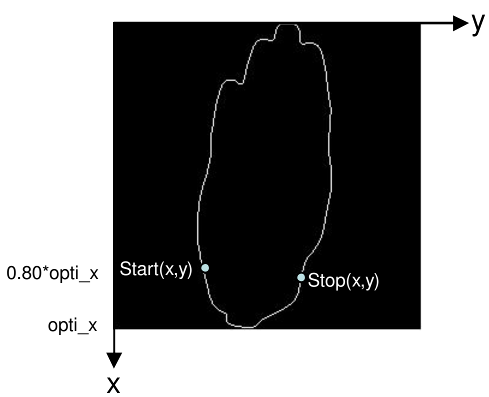

Fig. 7 Tracing Starting & Ending Point of Hand Image

This is our starting point to trace boundary and ending point is starting point y position plus one i.e. next row of starting point where white pixel exist.

Condition I: we start with UP=1, we first travel to top and check whether white pixel exist or not. If exist then continue in same way if not we check it on top left or top right. Again we search on top side and continue until we don't get any pixel on top or top-left or top-right. Condition I is demonstrated using F**i**gure 8.

0	**1**	0	0	0	0	0
0	**1**	**0**	0	0	0	0
0	0	**1**	0	0	0	0
0	0	**1**	0	0	0	0
0	0	**1**	0	0	0	0
0	0	**1**	0	0	0	0
0	0	**1**	**0**	0	0	0
0	0	0	**1**	0	0	0
0	0	0	**1**	0	0	0

0	0	0	0	0	**1**	0
0	0	0	0	0	**1**	0
0	0	0	0	**0**	**1**	0
0	0	0	0	**1**	0	0
0	0	0	**0**	**1**	0	0
0	0	0	**1**	0	0	0
0	0	0	**1**	0	0	0
0	0	0	**1**	0	0	0
0	0	0	**1**	0	0	0

Fig. 8 Condition I

Condition II: If we don't get any pixel it means we have to search on existing pixels right side, if pixel exist we follows the same way until we get no pixel on right side. We again follows as per condition I. if condition I and II not satisfied it means we have to search down, here we mark as peak as shown in figure 9.

0	0	0	0	0	0	0	0	0
0	0	0	**0**	**1**	**1**	**1**	0	0
0	0	**0**	**1**	0	0	0	0	0
0	0	**1**	0	0	0	0	0	0
0	0	**1**	0	0	0	0	0	0
0	**0**	**1**	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0

Fig. 9 Condition II

If condition I and II not satisfied then we search on down side by making DN=1

Condition III: we start with **DN=1,** We first travel to down and check whether white pixel exist or not. If exist then continue in same way if not we check it on down left or down right. Again we search on down side and continue until we don't get any pixel on down or down -left or down -right.

0	0	0	0	0	0
0	0	0	0	0	0
1	**1**	**1**	**1**	0	0
0	0	0	**0**	**1**	0
0	0	0	0	**0**	**1**
0	0	0	0	0	0

0	0	0	0	0	**1**
0	0	0	0	**0**	**1**
0	0	0	0	**1**	0
0	0	0	**0**	**1**	0
0	0	0	**1**	0	0
0	0	0	**1**	0	0

Fig. 10 Condition III

Condition IV: If we don't get any pixel it means we have to search on existing pixels right side, if pixel exist we follows the same way until we get no pixel on right side and then we follows condition III.

If in condition IV there is no pixel on right side we search on existing pixels left side, if pixel exist we follows the same way until we get no pixel on left side and then we follows condition III.

0	0	0	0	0	0	**1**	**1**
0	0	0	0	0	**1**	**0**	0
0	0	0	**1**	**1**	**0**	0	0
0	**1**	**1**	**0**	0	0	0	0
1	**0**	0	0	0	0	0	0

0	**1**	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0
0	**0**	**1**	0	0	0	1	1
0	0	**0**	**1**	**1**	**1**	0	0
0	0	0	0	0	0	0	0

Fig. 11 Condition IV

If condition III and IV not satisfied it means we have to search on top side, here we mark as valley.
After marking valley we again start from condition I. This way we keep on tracing peaks and valleys until we reach at stop point as shown in figure 12.

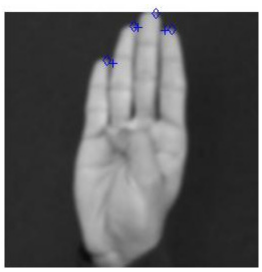

Fig. 12 Marking of Peaks and valleys

2.4 Feature Extraction

Image is then divided in to 16 parts, each of size 16*16 and naming them as A1, A2…A16. We then count number of peaks and number of valleys in image as shown in figure 13.

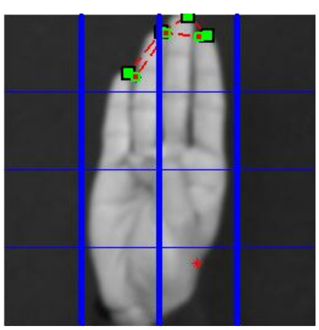

Fig.13 Image divided in 16 parts

From the divided image we find other parameters like in which part the highest peak has been detected in an image and which areas have been occupied by peaks and valleys.

Using these parameters a Neural Network is trained. For Neural Network training we have collected data base of 20 persons for the signs shown below in figure 14.

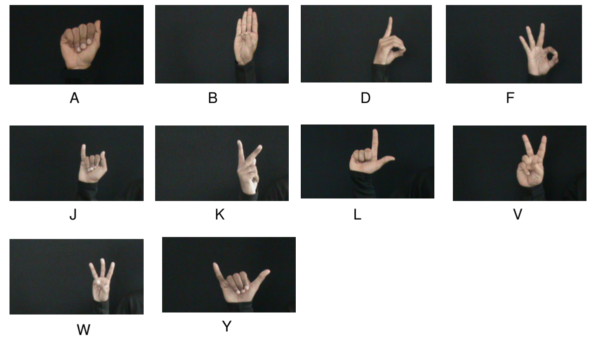

Fig. 14 American Sign Language Gestures

3. Recognition of sign using Neural Network

The Support Vector Machine (SVM) is used for classification. The parameters that we have set are as follows.
Data for training: 100%
Data for testing: 20%
Input PE's:50
Output PE's:10
Exemplars: 180
Hidden layer: 0
Step size: 0.01
Epochs: 1000
Termination-incremental: 0.0001
No. of Runs: 3

A result for training and testing database is shown in Table-1 and Table 2.

Table 1: Result on Training Data set.

Output / Desired	A	B	D	F	J	K	L	V	W	Y
A	18	0	0	0	0	0	0	0	0	0
B	0	19	0	0	0	0	0	0	0	0
D	0	0	18	0	0	0	0	0	0	0
F	0	0	0	18	0	0	0	0	0	0
J	0	0	0	0	18	0	0	0	0	0
K	0	0	0	0	0	17	0	0	0	0
L	0	0	0	0	0	0	19	0	0	0
V	0	0	0	0	0	0	0	18	0	0
W	0	0	0	0	0	0	0	0	17	0
Y	0	0	0	0	0	0	0	0	0	18
Result(%)	100	100	100	100	100	100	100	100	100	100

Table 2: Result on Testing Data set.

Output / Desired	A	B	D	F	J	K	L	V	W	Y
A	2	0	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0	0
D	0	0	2	0	0	0	0	0	0	0
F	0	0	0	2	0	0	0	0	0	0
J	0	0	0	0	2	0	0	0	0	0
K	0	0	0	0	0	3	0	0	0	0
L	0	0	0	0	0	0	1	0	0	0
V	0	0	0	0	0	0	0	2	0	0
W	0	0	0	0	0	0	0	0	3	0
Y	0	0	0	0	0	0	0	0	0	2
Result (%)	100	100	100	100	100	100	100	100	100	100

4. Conclusion

Detecting peaks and valleys algorithm is simple and easy to implement to recognize signs which belong to American Sign Language. For recognition we have extracted simple features form images and network is trained using Support Vector Machine. The accuracy obtained in this work is 100 % as only few signs have been considered here for recognition. In future work authors will try to recognize all signs of American Sign Language including dynamic signs which involves hand motion and design system which will convert signs into text or spoken words.

References

[1] Ravikiran J. et.al., “Finger Detection for Sign Language Recognition”, Proceeding of International MultiConference of Engineering and computer Scientists, 2009, Vol.1.

[2] Iwan Njoto Sandjaja, Nelson Marcos, “Sign Language Number Recognition”, proceeding of 5th International Joint Conference on INC,IMS and IDC, 2009, pp. 1503-1508.

[3] Jianjie Zhang, Hao Lin, Mingguo Zhao, “A Fast Algorithm for Hand Gesture Recognition Using Relief”, proceeding of 6th International Conference on Fuzzy Systems and Knowledge Discovery, 2009, Vol.1, pp. 8-12.

[4] V.Radha, V.Radha, “Threshold based Segmentation using median filter for Sign language recognition system”, proceeding of World Congress on Nature & Biologically Inspired Computing, 2009, pp. 1394-1399.

[5] Ryszard S. Chora´s Institute of Telecommunications, “Hand Shape and Hand Gesture Recognition”, IEEE Symposium on Industrial Electronics and Applications, October 4-6, 2009, pp.145-149.

[6] Salma Begum, Md. Hasanuzzaman, “Computer Vision-based Bangladeshi Sign Language Recognition System”, Proceedings of 12th International Conference on Computer and Information Technology, 21-23 Dec. 2009, pp. 414- 419.

[7] Yang quan, Peng Jinye, Li Yulong, “Chinese Sign Language Recognition Based on Gray-Level Co-Occurrence Matrix and Other Multi-features Fusion”, 4th IEEE conference on Industrial Electronics & Application, 2009, pp. 1569-1572.

[8] Yang Quan, Peng Jinye, “Chinese Sign Language Recognition for a Vision-Based Multi-feature Classifier”, International Symposium on Computer Science and Computational Technology, 2008, pp.194-197.

[9] Paulraj M P et.al., “Extraction of Head and Hand Gesture Features for Recognition of Sign Language”, International Conference on Electronic Design, 2008, pp. 1-6.

[10] Rini Akmeliawatil et.al., “Real-Time Malaysian Sig Language Translation using Colour Segmentation and Neural Network”, Instrumentation and Measurement Technology Conference Proceeding, 2007, pp. 1-6.

[11] Nilanjan Dey, Anamitra Bardhan Roy, Moumita Pal, Achintya Das ,” FCM Based Blood Vessel Segmentation Method for Retinal Images”, ijcsn, vol 1, issue 3,2012

[12] Tan Tian Swee et.al., “Wireless Data Gloves Malay Sign Language Recognition System”, 6th International Conference on Information, Communication & Signal Processing, 2007, pp. 1-4.

[13] Maryam Pahlevanzadeh, Mansour Vafadoost, Majid Shahnazi, “Sign Language Recognition”, 9th International Symposium on Signal Processing and Its Application, 2007, pp. 1-4.

[14] M. Mohandes, S. I. Quadri, M. Deriche, “Arabic Sign Language Recognition an Image–Based Approach”, 21st International Conference on Advanced Information Networking and Applications Workshops, 2007, pp. 272-276.

[15] Qi Wang et.al., “Viewpoint Invariant Sign Language Recognition” 18th International Conference on Pattern Recognition, 2005, pp. 456-459.

[16] Eun-Jung Holden, Gareth Lee, and Robyn Owens, “Automatic Recognition of Colloquial Australian Sign Language” Proceedings of the IEEE Workshop on Motion and Video Computing, 2005, pp. 183-188.

[17] Tan Tian Swee et.al., “Malay Sign Language Gesture Recognition System” , International Conference on Intelligent and Advanced Systems, 2007, pp. 982-985.