International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Hand Gesture Recognition Using Image Analysis and Neural Network

Ram Shankar Kumar[1], Dr. Nishant Srivastava[2]
Computer Sci- ence and Engineering department
Jaypee University Anoopshahr, Patna, India
Tel.:9893300543,Tel.:8709174822
ramshankarkumar07@gmail.com[1], nishant.srivastava@mail.jaypeeu.ac.in[2]

Abstract—Gesture Recognition is one of the most popular and viable solution for improving Human Computer Interaction. It has become very popular in the recent years due to its use in gaming devices like X box, PS4 and other devices like laptops, smartphones etc. Key issues in the hand gesture recognition system are removal of noise and complex background from the images. In the present study image are pre-processed using techniques like image enhancement and segmentation to extract the hand image without background. Gestures are then identified by finding number of defects in the convex hull of the hand. For speeding up the process a multi layer neural network is trained using back-propagation algorithm to classify different hand gestures. Experimental results have shown that the proposed system can recognize 5 classes of hand gesture with an accuracy of 94% after proper training. Performance of the system is found to be comparable with many state-of-the-art systems available today.

Keywords: Gesture recognition, biologically inspired com- puter vision, preprocessing, segmentation, convex-hull, convexity- defect, neural network.

1. INTRODUCTION

Hand gestures recognition falls into the categories of HCI that is human computer interaction. There are two approaches which are divided into Data-Glove based and Vision based approaches [1]. These two approaches are basically different due to the different nature of the sensory data collected. The Data-Glove method collects the data received from the sensor which are attached to user's hand. Using this methodology only this information are gathered which are important for the recognition system. Because of gathering of only necessary data it minimizes the cost of pre-processing and needless data. However, in real life scenarios use of a Data-Glove is often infeasible and can present different issues like connectivity, sensor sensitivity and a good deal of other hardware related problems [2]. On the other hand vision based approaches offer the convenience of hardware simplicity. This method only requires a camera or some sort of scanner. This type of approach refers to the biological human vision by artificially describing the visual field. While in term of the cost this type of approach is way cheaper than its Data-Glove counterpart, it generates a large body of data that is needed to be carefully processed in order to get only the important and necessary information. Keeping this in mind that to tackle, this recog- nition system needs not to be sensitive to lighting conditions, background invariant and also subject and camera independent [3]. Also, the challenging part of the hand gesture recognition problem is the fact that these systems need to provide real- time interaction with the system and the users. While this should not affect the model training directly it implies that later classification needs to be conducted in a manner of very small fraction of second. In recent years deep learning has come with a great result when it comes to computer vision problems. Deep learning approaches have shown to be highly accurate in various computer vision challenges on multiple topics. This increases the performance of these models is partially due to the recent advancement in technology related to world of GPU design and architectures. GPU are parallel in nature and are especially well-adjusted for training these types of models in very less time as compared to CPU. Among many neural network architecture research has proved that Convolution Neural Networks are most reliable and applicable to computer vision problems. As we know that humans are having the most sophisticated and reliable vision system, which similarly to convolutional neural networks consists of hierarchically distributed layers of neurons which act as processing units. Parameter sharing between neurons from different levels in the structure yield different connection patterns with different connection weights, which in turn concludes the process with classification.

2. SYSTEM COMPONENTS

A hand gesture recognition system having low cost and it can be run on any devices which are having their inbuilt camera or having external camera with them. This recognition system should work on different lighting conditions and having different backgrounds. Fig 1 shows an overview of our hand gesture detection and recognition framework.

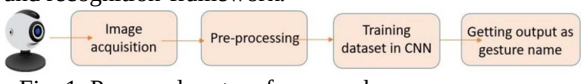

Fig. 1. Proposed system framework

Our proposed work depends on the following approach:

1) **Acquisition:** An image from the webcam is captured.
2) **preprocessing:** The image is detected and after that preprocessing is done on the Image.
3) **Training:** After pre-processing of the data, neural net- work is trained by those pre-processed data.
4) **Getting output:** After training classification is done on the data and output is given on the computer screen.

3. DATABASE DESCRIPTION

We have created hand gesture database by using laptop webcam which consist of 10 counting hand gestures. We give test images as input through the inbuilt camera to the system and system tells us our hand gestures. The system fully depend on data. We take gray scale image for making segmentation problem easy. We have created 5 classes for 5 different types of gestures as shown in the Fig 2 and Fig 3.

Fig. 2. Sample images from data set

Fig. 3. Sample images from data set

Each class of training set is having 2000 images. Each class of test set is having 500 images. Dimension of each image is like width is 100 pixel and height is 89 pixels.

4. PRE-PROCESSING

Pre-processing is a very important and necessary required task to be done in hand gesture recognition system. In our data set we have taken total 5 class and each class is having 2000 images. Pre-processing is applied to images before we can extract features from hand images. Pre-processing consist of two stages:

1) Segmentation of the images.
2) Morphological filtering on the images.

In the process of pre-processing of the images the first step is segmentation on the image which helps in converting a gray scale image into a binary image so that when the computer sees that image it can easily recognize the extracted hand from background in binary image. In our proposed work we have implemented the algorithm that is known as Otsu algorithm which is used in process of segmentation. After the conversion of grayscale image into the binary image we check for the noise in that binary image, so we use morphological filter technique. Morphological techniques consist of operations like: dilation and erosion.

A. Segmentation

To select a satisfactory level of threshold of gray level a very good and precise segmentation is needed for extracting hand gestures from different backgrounds. It is kept in mind that no background should have hand gesture part and no hand gesture part have any background image part. One chooses the segmentation process according the task and work that has to be done by the image. According to a test applied algorithm was found to give good segmentation results for gestures recognition and was, therefore, selected. This algorithm is non- parametric and unsupervised method of automatic threshold selection. [5]

B. Morphological filtering

After doing the process of segmentation on the image we may sometimes see noise in the image which is nothing but the unwanted pixels in the image. Background of the image may have some binary number which shows 1 called as background noise and hand gesture part may have some 0 that is called as gesture noise. Error of these types can create problems in process of contour detection of hand gesture, so we strictly need to remove these errors. So a method called as morphological filtering is applied on the image using dilation and erosion process to get a clear picture of our hand gesture. [5]

In the morphological dilation and erosion we implement some rules on a binary image which has been converted from grayscale image into binary image. Any given pixel's value in output image is obtained by allying set of rules on the neighbors in the input image.

1) Dilation: Dilation is the basic operators, when it comes in the field of morphological process. It is mainly applied to binary image but there are many versions of dilation which works on gray scale images also. The function of this operator on a binary image is to enlarge the boundaries of regions of foreground pixels gradually. In this process foreground pixel areas increase while holes within this area becomes smaller. Fig4 shows the effect of dilation using a 3 x 3 square matrix structuring elements. In fig4 foreground pixels are represented by 1s and background pixels are represented by 0s.

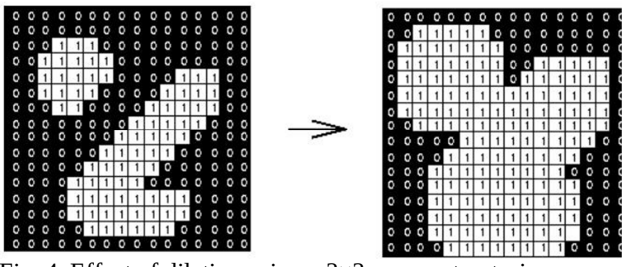

Fig. 4. Effect of dilation using a 3×3 square structuring element

2) Erosion: Erosion is also the basic operators, when it comes in the field of morphological process. The function of this operator on a binary image is to erode away the boundaries of regions of foreground pixels. In this process areas of foreground pixels shrink in size while holes within those areas become larger. It is just opposite to dilation process and dilation process is just opposite of erosion process. Fig5 shows the effect of erosion using a 3 x 3 square matrix structuring elements. In fig4 foreground pixels are represented by 1s and background pixels are represented by 0s.

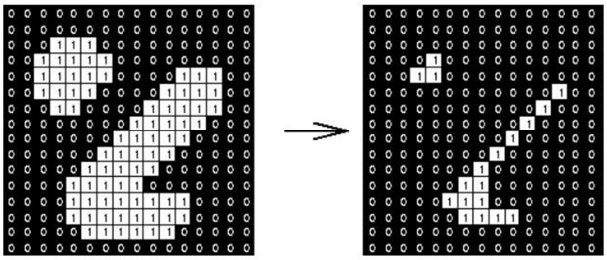

Fig. 5. Effect of dilation using a 3×3 square structuring element

5. PROPOSED WORK

In this study we have used two methods to detect hand gestures:

1) By using convexity defect in convex hull
2) By training the convolutional neural network

A. Method 1

In this method we have done image analysis and have found convexity defects in the image to detect the gesture of the image. To find the convexity defects in the hand image first we find contour of the hand image and convex of the hand image.[7]

1) Contour detection: This process is done after the pre- processing of the image. When a boundary of an image is having continuous points and those points are having same color and same intensity then a curve is made by joining all its boundary points and that curve is known as the contour of that image. Finding contour of an image can be a useful process for analysis of image shapes and object detection.

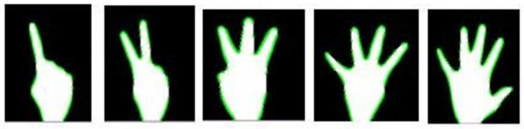

Fig. 6. Sample output of contour detection

*2) **CONVEX HULL:*** A line completely surrounding a set of points in a plane without having any concavities in the line is known as the convex hull. Or in other word, we can explain it like smallest polygon having which is having all the points inside it is known as convex hull. [8]

*a) **Gift wrap algorithm for finding convex hull:*** As we know that convex hull is very essential process in pre- processing of the image, and at the same time it is also helpful in constructing many other structures and unsupervised image analysis.

We can easily think about the visualization of convex hull by a sort example. Suppose that there are many nails pointed on a plane surface. Now take an elastic rubber and make a stretch around the nails and leave it. After leaving the rubber band it will snap around the nails. Whichever nails touches the elastic rubber band and whatever the area is made by that rubber band by touching nails is known as the convex hull.

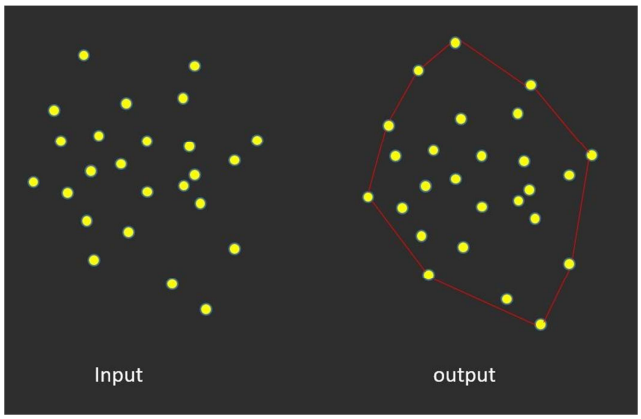

Fig. 7. Example of convex hull

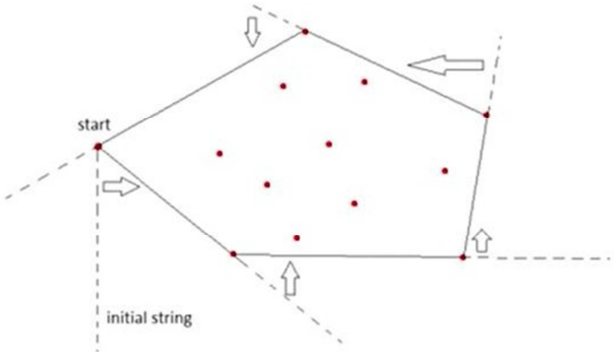

Fig. 8. Visualization of gift wrap algorithm

In field of geometry of computation, this algorithm is

for finding the convex hull of a given set of points. In the case of 2-D this algorithm is given a name called Jarvis march after R. A. Jarvis, who had published it in 1973. It has the time complexity of O(nh), here n is the number of points and h is the number of points on the convex hull.

b) ***Convexity defect****:* When any object is segmented out from an image there is a cavity in that object that is known as the convexity defect. It means the area which is not belonging to that object but inside outer boundary of the convex hull it is located. [8]

After finding the convexity defects of the hand we applied one condition in our convexity defects feature that if the defects angle is less than 90 degree than only the defect will be accepted.We applied this condition because every finger gap of a human is not more than 90 degree so this feature will be valid for all type of hands smaller or bigger.

But by this method we were not able to solve the gesture classification problem other than those five 5 classes as we were not getting the convexity defects for all the classes. We got the results only for the class One Two, Three, Four and Five but not for the gestures other than One, Two, Three, Four and Five.

So we tried to take our proposed work to the next level so that the solution of gesture recognition problem can become

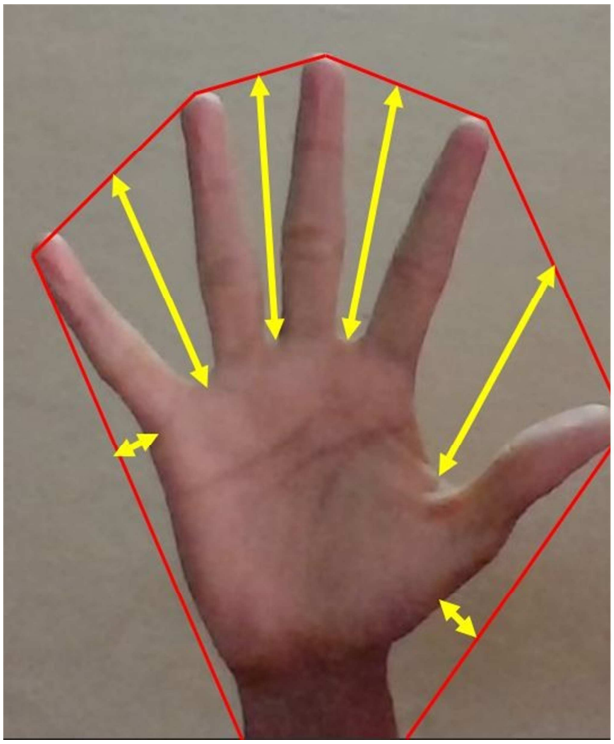

Fig. 9. Image of hand gesture having convexity defect

Fig. 10. Sample output of convexity defect

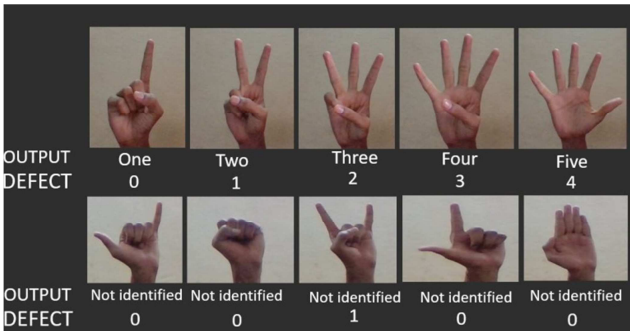

Fig. 11. Output data from method 1

flexible. We increased our classes by adding new gestures which had not been recognized by the convexity defect. And thus here comes the convolutional neural network in the picture. So to make our project flexible and full-fledged recognition system for all the classes we have used the convolution neural network.

B. Method 2

In this method instead of finding convexity defect of the image we have used convolutional neural network for the feature extraction and classification. Convolutional neural net- works are reliable and powerful image processing artificial intelligence that use deep learning process to perform both type of tasks like generative and descriptive tasks. It also uses machine vision that includes image recognition and video recognition. A convolutional neural network also uses a system which is just like a multi layer perception which has been designed for reducing image processing tasks. The layers of a convolutional neural network mainly consists of three type of layers that is an input layer, an output layer and a hidden layer which includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers.

Most important steps for building a CNN is:

1) Convolution[9]
2) Pooling[9]
3) Flattening

1) Convolution: The main aim of convolution is to extract and get important features from the given input images. Convolution keeps the spatial relationship between pixels by getting metadata from image features using small squares of input data. After every convolution operation an additional operation is used which is called as ReLU and it is also denoted by activation function. As shown in fig12 consider a 7 x 7 image whose pixel values are only 0 and 1. And consider another 3 x 3 matrices.

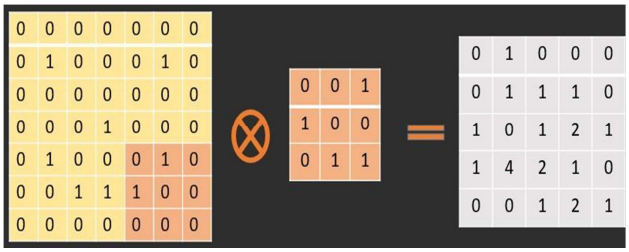

Fig. 12. Process of convolution

2) Pooling: After convolution a pooling operation is done and it is also known as sub sampling or down sampling which reduces the dimension of each feature map but extract the most important information. It is of two type, and they are max pooling and average pooling. In process of max pooling we generally define a spatial neighborhood for example, a 2 2 window and choose the largest element from the rectified feature map within that window. And in process of average pooling Instead of taking the largest element we take average or sum of all the elements. Fig 13 shows the example of max Pooling.

3) Flattening: Flattening is the process which comes after the process of pooling. In this process a matrix is converted into a straight linear array so that we can input it into the nodes of our neural network.

So finally a CNN network looks like this as shown in fig14: Our model was originally designed for pixel based segmen- tation of images. Since the process of segmentation essentially rests on pure classification, small tuning to the kernel and filter sizes of the architecture allowed for this architecture to achieve relatively good performance in this task. The algorithm that has been used for training the model is Back-propogation

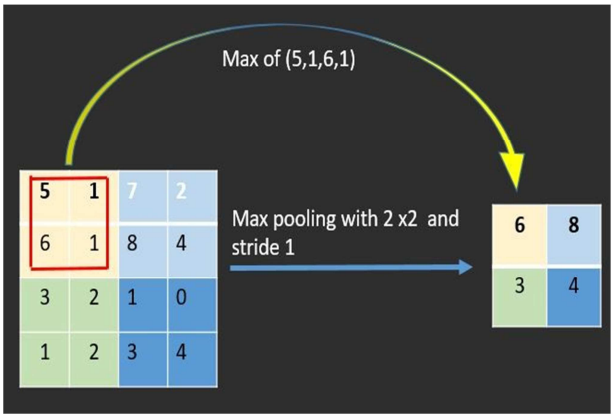

Fig. 13. Rectified feature map in max pooling operaion

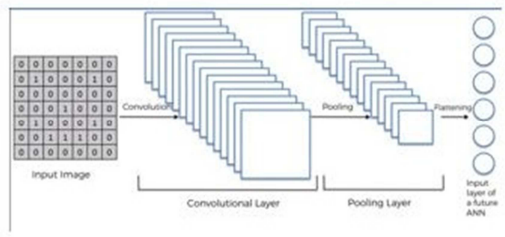

Fig. 14. A convolutional neural network

[10]. Our model has 14 layers that contain 5 convolutional and 5 pooling layers [9]. Before the softmax classifier there is a fully connected layer aggregating the convolved features generated to this point. Finally, the input layer creates a 100x 89px receptive field.

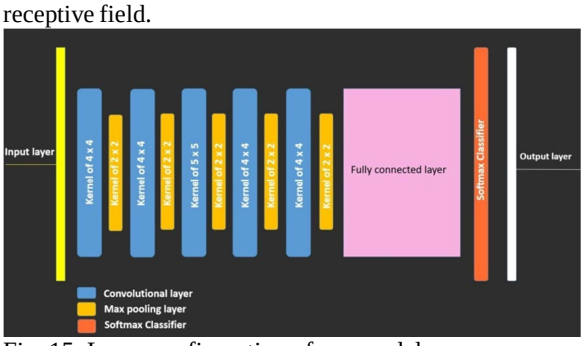

Fig. 15. Layer configuration of our model

This model was implemented in Keras framework using the Tensorflow as back end with Python language. Each of the neurons contained in this network relies on the Recti- fied Linear Unit activation introduced in [11]. In this model classification is performed using a Softmax classifier with 5 output neurons one for each class. Fig15 shows the models' configuration layer by layer with unit numbers and kernel sizes. As with the previous models, this model was trained no more than 20 epochs in its best run. For increased performance we started training this model with an initial learning rate of 0.001.

6. TRAINING PHASE

In training part of the convolutional neural network, the weight matrices between the input and the hidden layers and output layers are initialized with any random values. The data that has come from the output of the neural network are compared with the desired values and then the error are computed. This pattern is repeated again and again, until the model detects the error rate of the output layer touches a minimum value. Again this process is then repeated for the next input value, until all the values of the input data have been processed. The activation function that has been used in our model is reLu. The training algorithm used is back- propagation.[10]First Features are extracted by network and it is entered as training data into the convolutional neural network. The result of classifier

directly depends on how well the input data has been entered. Fig 16 shows the train accuracy and validation accuracy graph and Fig 17 shows the train loss and validation loss graph.

7. TESTING PHASE

After training phase we come to last phase that is testing of the model. In this phase, we extract the features in the same manner as we have extracted in the training phase. In testing phase, we are having 2500 hand gesture images which are used to test the proposed system. All the testing images are divided into 5 classes.

8. RESULTS AND DISCUSSION

In this particular research, it was important to evaluate performance in both accuracy and operation timing due to the potential of applying this types of models in a real-time control scenario. The results of the experiment is presented to show the effectiveness of the system. Our hand gesture recognition system was carried out on a 2.20GHz Intel® Core™ i7-8750 CPU, 8 GB RAM on windows 10 platform. We have achieved 94% recognition rate with our captured data.

9. CONCLUSION

In recent year lot of research has been conducted in gesture recognition field. The main purpose of the project was to build a Hand Gesture recognition system. We have shown in this project that hand gesture recognition system can be designed using the convexity defect but it is not the up to the mark solution for problem like gesture recognition. So to speed up our project and to make reliable solution for gesture recognition problem we have used convolutional neural network. And finally our proposed work gives 94% accuracy in detecting the hand gestures taken from a webcam. We have also compared our network with the VGG model also known as Visual Geometric Group model. We have trained that model on the same data that we have used for training our model, and we have found that VGG model accuracy is 96%. Thus, we can say that difference of accuracy between our model and VGG model is acceptable.

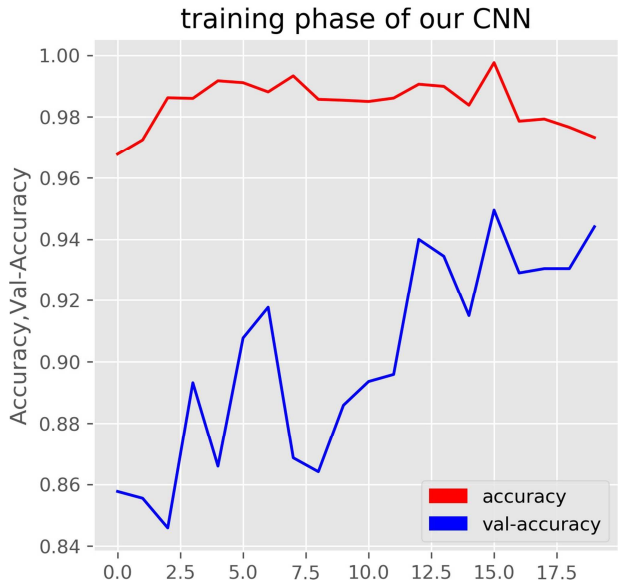

Fig. 16. Accuracy and validation Accuracy graph

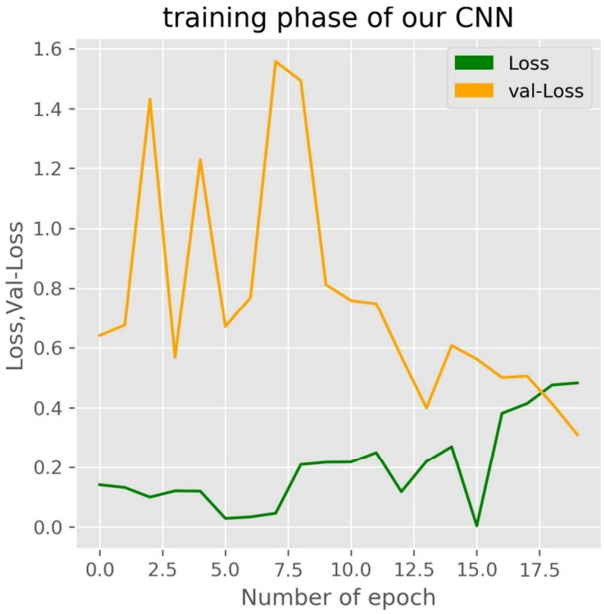

Fig. 17. Loss and validation loss graph

Fig. 18. Final output of our proposed work

REFERENCE

[1] Shastry, K.R., Ravindran, M., Srikanth, M., Laksh- mikhanth, N., et al.: Survey on various gesture recognition techniques for interfacing machines based on ambient intelli- gence. arXiv preprint arXiv:1012.0084 (2010)
[2] Singer, M.A., Goldin-Meadow, S.: Children learn

when their teacher's gestures and speech differ. Psychological Sci- ence 16(2) (2005) 85–89

[3] Ohn-Bar, E., Trivedi, M.M.: Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. Intelligent Transportation Systems,IEEE Transactions on 15(6) (2014) 2368–2377

[4] Strezoski, G., Stojanovski, D., Dimitrovski, I., Madjarov, G.: Content based image retrieval for large medical image cor- pus. In: Hybrid Artificial Intelligent Systems.Springer (2015) 714–725

[5] E. R. Dougherty, "An Introduction to Morphological Image Processing", Bellingham,Washington: SPIE Optical En- gineering Press, 1992.

[6] L. Gupta and T. Sortrakul, "A Gaussian mixture based image segmentation algorithm," Pattern Recognit., vol. 31, no. 3, pp. 315–325, 1998.

[7] Lalit Gupta and Suwei Ma, "Gesture-Based Interac- tion and Communication: Automated Classification of Hand Gesture Contours," IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 31, no. 1, February 2001

[8] R. K. Cope and P. I. Rockett, "Efficacy of gaussian smoothing in Canny edge detector," Electron. Lett., vol. 36, pp. 1615–1616, 2000

[9] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(1),221–231 (2013)

[10] Dr. Rama Kishore, Taranjit Kaur: Backpropagation Al- gorithm: An Artificial Neural Network Approach for Pattern Recognition.International Journal of Scientific & Engineering Research, Volume 3, Issue 6, June-2012

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems.(2012) 1097–1105