ISSN 2229-5518

Neural Network based finger counting technique

Prof.Ms.M.N.Bansode, Prof.S.D.Jambhale Dr. S. K. Dixit

Abstract-A primary goal of hand gesture recognition research is to create a system which can identify specific human hand gestures and use them to convey information or for device control. Gestures are physical positions or movements of a person's fingers, hands, arms or body used to convey information. Gesture recognition is the process by which gestures formed by a user are made known to the system. There are different methods of Gesture Recognition. Here we are using Neural Network for hand Gesture recognition. When speaking about image recognition or sign classification, the most widespread solution is the neural network. A program can be developed in MATLAB for neural network for recognizing the number of the fingers in front of web camera. It's a highly efficient method that has been proven able to distinguish and classify with an amazing rate of performances. Then, it will be possible to train the network with a full set of examples and finally to use in real time conditions.

Keywords: Hand gesture recognition, image processing, neural network, Mat lab & neural network.

I. INTRODUCTION

Since the introduction of the most common input computer devices not a lot have changed. This is probably because the existing devices are adequate. It is also now that computers have been so tightly integrated with everyday life, that new applications and hardware are constantly introduced. The means of communicating with computers at the moment are limited to keyboards, mice, light pen, trackball, keypads etc. These devices have grown to be familiar but inherently limit the speed and naturalness with which we interact with the computer.
In recent years there has been a great deal of studies aimed at the inconvenience of human computer intercommunication tools such as keyboard & mouse. As one of the alternative gesture recognition methods have been developed by which a variety of commands can be used naturally. Since conventional input devices need a great deal of technical education, many researches feel a great interest in & attach importance to hand gesture recognition. In the present day, framework of interactive, intelligent computing, an efficient human –computer interaction is assuming utmost importance in our daily life.

Gestures are physical positions or movements of a person's fingers, hands, arms or body used to convey information. Hand gestures, i.e., gestures performed by hand. Gesture recognition is the process by which gestures formed by a user are made known to the system.

A B C

A pattern recognition system will be using a transform that converts an image into a feature vector, which will then be compared with the feature vectors of a training set of gestures. The final system will be implemented with a neural network.

II. LITERATURE REVIEW

In recent years there have been great deals of studies aimed at the inconvenience of human computer intercommunication tools such as keyboard & mouse. As one of the alternative gesture recognition methods have been developed by which a variety of commands can be used naturally. Since conventional input devices need a great deal of technical education, many researches feel a great interest in & attach importance to hand gesture recognition. In the present day, framework of interactive, intelligent computing, an efficient human –computer interaction is assuming utmost importance in our daily life. Gesture recognition can be termed as an approach in this direction [1]. The major tools used for this purpose includes HMMs [2] & ANNs [3].HMMs tool basically deals with the dynamic aspects of the gestures. Gestures are extracted from a sequence of video images. But the major disadvantage of HMMs is that it is based on probabilistic framework [4]. For large data sets ANNs have been used for representing & learning the gesture information. ANNs is mostly used for recognizing a static posture.
Gesture recognition has a wide range of applications. It finds it suitability for the peoples with physical disabilities [4]. It enables very young children to interact with computers. It also helps for recognizing sign language [5]. It can be also effectively used in communicating in video conferencing [6]. Limited set of hand gestures from hand images can be used for controlling the robot motions [7].

III. OBJECTIVE OF WORK

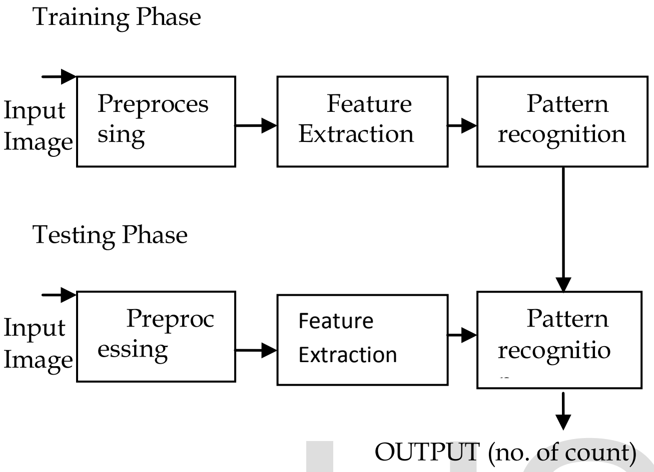
The goal of this project is to develop a program implementing gesture recognition. At any time, a user can exhibit his hand doing a specific gesture in front of a video camera linked to a computer.

ISSN 2229-5518

The program has to collect pictures of this gesture, analyze it and to identify the sign. In order to lighten the project, it has been decided that the identification would consist in counting the number of fingers that are shown by the user in the input picture.

IV. CONCEPTUAL BLOCK DIAGRAM & PROCEDURE

The Conceptual block diagram of the work is shown below.



PROCEDURE:

a) Image creation: The images are taken by database.

b) Preprocessing: It is used to filter out noise from the image. Image captured by the camera contains noise like impulse noise & Salt-pepper type noise. Salt –pepper noise is one of the types of the saturated impulse noise. This type of noise is removed by using Median filter.

c) Feature Extraction:

In this step the required features are extracted from the image. The following steps are used:

1. Subtraction: In background subtraction method, the image of background is captured as a reference image. The plain background image is subtracted .from the image having hand with background that is the Target image. The resulting image is only hand. This is the simple & easiest method of extraction.

2. Segmentation: Thresholding is used for clarity. Thresholding by Hysteresis method is used. The hysteresis will give the information regarding which point is to select as the threshold. Also it will convert the image into the binary form. So that it will be easy for further processing.

3. Thinning: Thinning is a operation that is used to remove selected foreground pixels from binary images. In this object is converted into the set of digital arcs. These arcs are lying roughly along the medial axis. That is it will give reduced amount of the data, reducing the time required for processing. The border pixel having more than one neighbour is removed, converting into the thin line. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output.

d) Pattern Recognition:

Creating Training sets: The pattern recognition block consists of creating the training sets from the images. Here the input & its expected output is known & according to that the Neural network is Trained, which calculates the weight & bias? This creates the Training sets.

It is possible to build different set:

A set of 50 examples: 10 examples for each sign

A set of 100 examples: 20 examples for each sign

A set of 150 examples: 30 examples for each sign

Testing: For testing, the unknown image is given as the input to network, which is then compared with the training set. Depending upon the match it gives the output.

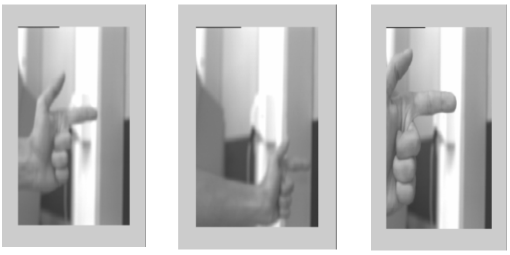
IV. DEFINING THE DIFFERENT ISSUES

Collecting the Pictures

First of all, and obviously, it will be necessary to collect pictures. There is a choice to do concerning the way we want to collect these pictures, given that it depends on how we implement the main program. Running in the MATLAB environment requires the pictures to be saved in memory and called back when running the program, because the Image Acquisition Toolbox is not available on the MATLAB version used for the design of the program.

Finding the hand

Now, let's suppose that a set of representative pictures is provided. We need then to analyze the picture, and to find the relevant part of the picture. Indeed the user will never put his hand in the same area of the picture. Here are given few examples of the same sign done in different areas, which have to lead to the same identification result, which should be '2':



Zooming on the hand

Principles: How to find a hand in a picture?

According to the requirements, the video camera is not supposed to move. This piece of information gives a huge advantage that allows simplifying the zooming process. Indeed, it implies that the background is more or less always the same. In all what follows, it will be supposed that in the picture, we can just find the hand and the background: no other object should be present. As a consequence, it will be impossible to exhibit simultaneously the two hands and expect the program to process them both. Few examples of allowed and forbidden pictures are given here:

NOTE: Please note that all correct sign are suppose to include thumb finger, which is used as a reference for zooming on the hand. The reasons of this constraint will be discussed and explain later

V. NEURAL NETWORKS.

When speaking about image recognition or sign classification, the most widespread solution is the neural network. It's a highly efficient method that has been proven able to distinguish and classify with an amazing rate of performances. There are few issues linked to neural network solutions. First of all, it will be necessary to choose a network, say: How many hidden layers? How many neurons per layer? Then, it will be required to choose some learning parameters, say: How many epochs? How many learning examples?

Then, it will be possible to train the network with a full set of examples and finally to use in real time conditions.

Weighted Averaging Analysis

In order to understand the basic idea that is discussed here, let's consider the differences and the common points between the methods that have already been introduced:

The Pixel Counting method and the Edges Counting method were some very simple solutions, but their problem was they were not efficient enough. Their advantage was their low-complexity level for the implementation, given that they were geometrical solutions.

The Neural Networks solution has been proven quite more efficient, but it requires training, and special management and processing of the binary picture. Moreover, when looking at the weights of the input layer, it appears that the neural network just realizes a kind of weighted averaging.

Hence, the motivation in this section is to try to realize weighed averaging by a simpler way.

Efficiency of the algorithm:

That is to say that the algorithm has to realize the following operations:

Calculate

WA=

$$\sum_{\text{column 15}}^{\text{column 25}} \left[\text{number}_{\text{edges(column)}} * \left(\sum_{\text{line 1}}^{\text{line 30}} \text{pixel(line, column)} \right) \right]$$

Estimate the number of fingers in the picture using:

If WA< 30 then
Number _of _fingers=1
If 30<WA< 30 then
Number _of_fingers=2
If 170<WA< 470 then
Number _of_fingers=3
If 470<WA< 30 then
Number _of_fingers=4
If WA< 30 then
Number _of_fingers=5

NOTE: In such calculus, the bound for the three and a fourth finger, for example, has been determined using 3*100*3 and 4*100*4, say the general equation is:

WA=100[No. of edges/2]

The consequence is that the distance between typical WA values (values of the weighted averaging) increases at an exponential rate, and that makes the classification less sensitive to errors. Indeed, in this case, the bound between two close possibilities is always large: for example it has been said that the typical WA when 5 fingers is (960+1600)/2=1280. An error can occur only if the calculated WA, which should be 1280, is under 930, the calculation error has to be bigger than 350. This can happen only if there are a lot of errors on the number of edges in each column and if the relative dimensions of the fingers are "strange": one finger very thick, and three fingers very thin and the thumb.

In order to understand the efficiency of this method, let's compare it to the bound that would have been considered in a simple pixel counting algorithm: for four fingers, the sum of the pixel will be about 3*60=180, and for five fingers, it would be equal to 4*60=240. The bound between 4 and 5 fingers would be (180+240)/2=210. An error on five fingers happens when less than 210 pixels are counted in the columns 15 to 25. The margin is: 240-210=30.

When comparing the error margins, it appears that without any weights, it is equal to 30, and that with weights chosen as number of edges in the column of the analyzed pixel, this margin tend to 350, so more than 10 times the previous margin! That's why this method is quite better the simple pixel counting one: different number of fingers lead to different ranges that are separated by very large spaces that only huge errors can get through, and such errors are not very frequent.

Without weights, confusion may occur when several fingers are exhibited (three, four or five fingers). The use of weights makes these confusion quite more rare because three four and five fingers pictures turn into WA values that are very distant one to the other.

Performances

The results are sensitively less satisfying than when using a neural network: about 93% of the signs are correctly classified, just thanks to geometrical processing of the picture! Foe the neural network, 100% of the training set was correctly classified and 96% of new inputs were also classified. Such results are quite good given that it does not use any training, any standard-deviation calculus or any save of any net-matrix. The processing is quite fast given that no sophisticated calculus is required when running the program.

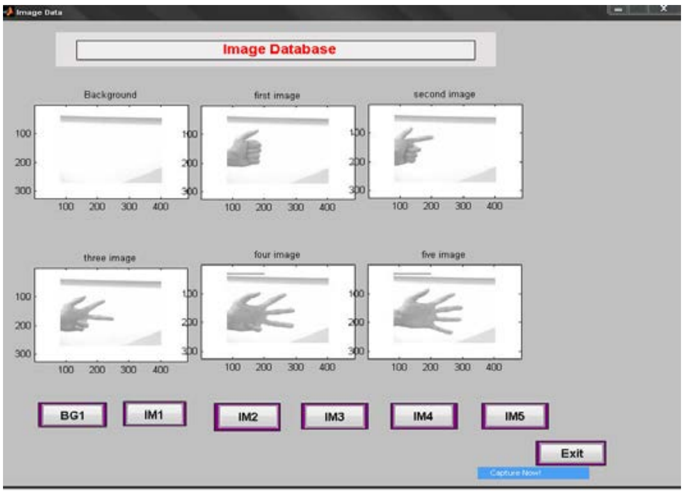
CONCLUSION:

As a conclusion, it appears that there are plenty of ways to solve such problems. If the neural network method is the most efficient solution when considering the rate of errors, another method based on attributing weights to each column of the input picture, is quite easier to develop and manipulate, and has also been proven very efficient.

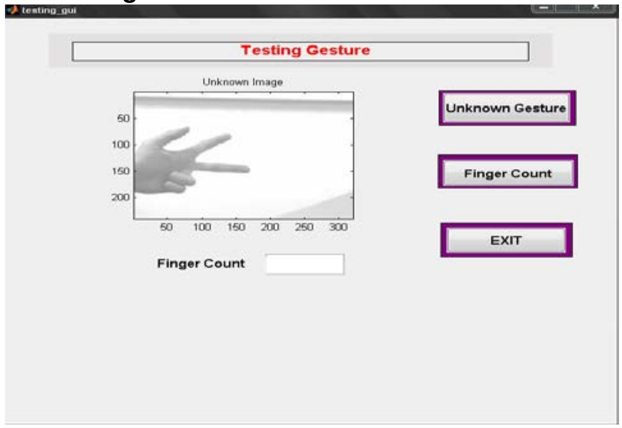
There are still some improvements to do, and a major problem to solve: find why does some Matlab files turned into files works perfectly when used in a Graphical User Interface, whereas other Matlab files turned into files using the same way lead to immediate run-time errors.

RESULTS:

Image database



Unknown gesture:



REFERENCES:

1. 1. Sushmita Mitra, Tinku Acharya, 2007," Gesture Recognition: A Survey", IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications And views,Vol. 37, No. 3, pp. 311-323.
2. 2. Sebastein Marcel, Oliver Bernier, 2005, " Hand gesture recognition using input-output Hidden Markov models", IEEE Transactions on Pattern recognition & Machine intelligence, Vol. 27, No. 8. pp-347-362.
3. 3. Yu Yuan, Ying Liu, Keneeth Barner, 2005, " Tactile gesture recognition for people with disabilities", International conference on Accoustic, speech & Signal Processing, Vol. 5,pp. 461-464.
4. 4. Kenji Oka, Yoichi Sato, Hideki Koike,2002, "Real-Time Fingertip Tracking and Gesture Recognition", IEEE trans. Computer Graphics & Applications, pp 64-71.
5. 5. Sylvie, C.W., Ong & Surendra Ranganath, 2005, "Automatic Sign Language analysis: A survey & the future beyond Lexical Meaning.", IEEE Transaction on Pattern recognition & Machine intelligence, Vol. 27, No. 6. Pp-873-891.
6. 6. Jose Antonio Monteno, Luis Enrique Sucar, 2006, " A decision theoretic video conference system based on gesture recognition", Proceedings of the 7th Intrnational conference on Automatic face & gesture recognition, Issue 10, pp. 387-392.
7. 7. Asanterabi Malima, Erol Özgür, and Müjdat Çetin, 2006, " A Fast Algorithm for vision based hand gesture recognition for robot control ", IEEE transaction on signal processing & communication Applications. pp.1 –
8. 8. Akio Ogihara, Hirishi Matsumoto & Akira Shiozaki, 2006, " Hand region extraction by background subtraction with renewable background for hand gesture recognition", IEEE International symposium on Intelligent signal processing & communication systems, pp. 227-230.
9. 9. Getting Started with MATLAB Version 6 □□ COPYRIGHT 19842001 by The Math Works, Inc.
10. 10. Image Processing Toolbox User's Guide □COPYRIGHT 1993- 2001 by The Math Works, Inc.