Computer System Department

Faculty of Computers and Information

Cairo, Egypt

Cairo University

Cairo University

Faculty of Computers and Information

Computer science (CS)

Graduation Project

2015

Mute-deaf gestures recognition

Team Name

(Team Number: 5)

ID	Name	Email	Department	Mobile
20110315	Mohamed Emad El-Dien Badawy	m.emad@stud.fci-cu.edu.eg	CS_IS	01272431975
20110341	Mahmoud Fekry Hassan	Eng.mahmoud7182@gmail.com	CS_IS	01148826189
20110312	Mohamed Atta Faheem	mohamedatta20110312@gmail.com	CS_IS	01142332862
20110486	Hazem Mahmoud Elshenawy	h.elshenawy93@gmail.com	CS_IS	01016331224
20120605	Mohammad Adnan Albukaai	adnanalbukaai@gmail.com	CS_IT	01159750246

Abstract

One of the most popular handicaps is the deaf and dumb type, which prevent person from listening and talking. The number of deaf and dumb in the world continuously increasing and they are introverted closed society. Therefore, Deaf-Dumb people do not have normal opportunities for learning. Uneducated Deaf- Dumb people face serious problem in communication with normal people in their society. It is notable, however, that most available application focus only on learning or recognition of sign language. In this paper, we introduce an integrated android application to blend uneducated Deaf-Dumb people within society, and help them to communicate with normal people. The introduced application proposes an easy translator in keyboard form that can translate some word from sign language to Arabic language and vice versa. This application also contains most daily words for teaching deaf and dumb kids in attractive way (colors, pictures, animations, quiz …etc). Moreover, it introduces some games that help them to communicate and entertain.

Acknowledgment

First of all, all praises to Allah for giving us the strength and His blessings in completing this work. Would like to thank DR : Amr Badr faculty of computer and information at Cairo university for supervising us in this project, also thanks go to TA: Mohammed Elarnaoty Faculty of Computers and information Sciences for helping us with him assistance in starting this project.

List of Figures:

Chapter 1

Introduction

1- Background:

Mute and deaf is a term means a person who could not either hear or both hear and speak [1]. The number of mute and deaf in the world continuously increasing and they are introverted closed society. The education of the deaf is only about one century old [2]. Since sign is the earliest way of communication in the world when there is no appropriate language, so the sign language is preferred among the deaf-dumb people for education. As with other forms of manual communication, Sign language depends on finger spelling. The simplest visual form of finger spelling is simulating the shape of letters in the air, or tactually, tracing letters on the hand. Finger spelling can use one hand such as in American Sign Language, French Sign Language and Irish Sign Language, or can use two hands such as in British Sign Language [3]. Uneducated mute and deaf people can communicate with other people (normal or handicaps) with sign language only, so they face serious problems in their daily life. For example: restaurants, transportation, hospitals, government offices...etc. Therefore, they need an effective tool to translate their words from sign language to Arabic or English language directly. This tool can facilities their communication with normal people and encourage them to learn both Arabic and languages. Also, Deaf-Dumb kids needs to learn sign, Arabic and English languages in an interesting way. For the above reasons, the motivation of our application is to offer a service to the society in general and to Deaf-Dumb people in particular. This work is an integrated system that can easily solve most of their problems in one application. Therefore, the present work aims to:

- Help deaf and dumb to interact more with normal people.

- Offer a great tool for parents to teach their deaf and dumb kids.
- Introduce Sign language keyboard.
- Introduce quizzes and games for training deaf and dumb to identify Arabic and English words.

Hoping this application can give a hand to uneducated Deaf-Dumb people who could not read and write Arabic languages to communicate with others, to learn and to entertain.

A- Deaf and mute: - people they cannot speak and listen the sound and deal with natural people with assign language. Deaf and mute is a person who can neither hear nor speak.

B- Gestures:

Figure 1 Hand gestures

Gestures are a form of nonverbal communication in which visible bodily actions are used to communicate important messages, either in place of speech or together and in parallel with spoken words. Gestures include movement of the hands, face, or other part of the body .A gesture in a sign language is a particular movement of the hands with a specific shape made out of them. A sign language usually provides sign for whole words. It can also provide sign for letters to perform words that don’t have corresponding sign in that sign language. It is very important to understand the meaning of gestures before you travel to different countries.

C- PCNN: (Pulse-Coupled Neural Networks) algorithm for Pattern Recognition that produced image segmentation. PCNNs have been utilized for a variety of image processing applications, including: image segmentation, feature generation, face extraction, motion detection, region growing, noise reduction, and so on. A PCNN is a two-dimensional neural network. Each neuron in the network corresponds to one pixel in an input image, receiving its corresponding pixel's color information (e.g. intensity) as an external stimulus. Each neuron also connects with its neighboring neurons, receiving local stimuli from them. The external and local stimuli are combined in an internal activation system, which accumulates the stimuli until it exceeds a dynamic threshold, resulting in a pulse output. Through iterative computation, PCNN neurons produce temporal series of pulse outputs. The temporal series of pulse outputs contain information of input images

and can be utilized for various image processing applications, such as image segmentation and feature generation.
Pulse coupled neural networks (PCNN) were introduced as a simple model for the

cortical neurons in the visual area of the cat's brain. Important research in the 80's and 90's led to the establishment of a general model for Such models proved to be highly applicable in the field of image processing, a series of optimal procedures being developed for contour detection and especially image segmentation.
The model proposed here is based on three modules of processing: the pulse-coupled neural network, the Discrete Fourier Transform (DFT) module and the multilayer perceptron (MLP) classifier.

The key of the entire system lies in the neural analyzer that, in our case, is made of pulse-coupled neurons, which act like local analyzer cells. The pulse train generated by the neurons is a direct result of stimulus excitation and lateral interaction between neurons. Lateral interaction and further stimulation determine the neurons to fire in synchrony in the homogenous areas associated to the image. These effects can be exploited in image segmentation. However, our assumption is that the pulse train of the neurons captures somehow morphological information from the image. The model we used for the network had been proposed by T. Lindblad and J.M. Kinser.

The pulse-coupled neuron is a particular type of leaky integrator neuron. The leak is modeled by the exponential terms in equations (1) and (2). The refractory period is simulated by increasing the threshold when the neuron fires and decreasing it exponentially after firing.

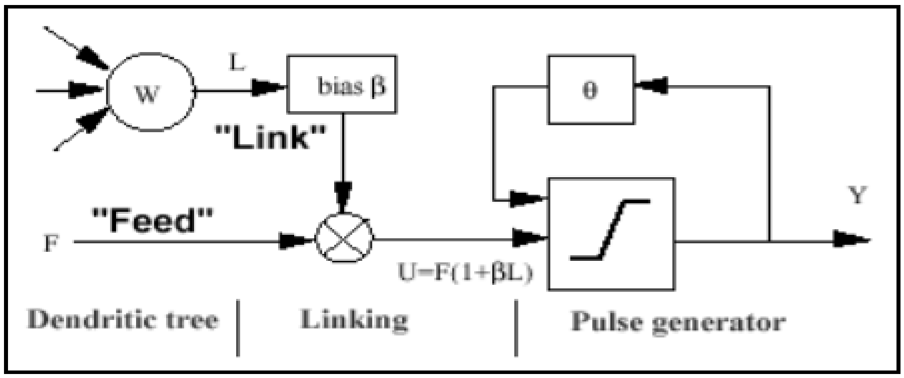


Figure 2 The structural model of the pulse-coupled neuron

In the next equations we will refer to “n” as being the current iteration (discrete time step)

Where "n" varies from 1to N-1(N - is the total number of iterations; n = 0 is the initial state). The dendritic tree can be described by the following equations:

$$Fij[n] = e^{-\alpha_F} \cdot F_{ij}[n-1] + V_F \cdot \sum_{kl} M_{kl} S_{ijkl} \tag{1}$$

$$Lij[n] = e^{-\alpha_L} \cdot L_{ij}[n-1] + V_L \cdot \sum_{kl} W_{kl} Y_{ijkl}[n-1] \tag{2}$$

The two main components F and L are called feeding and linking.
The (i,j) pair stands for
the position of the neuron in the map. α_F and α_L are time constants for feed and link. Sijkl
is the stimulus component computed from the pixel intensity (<i+k, j+l>, "<x,y> meaning the intensity of the pixel with coordinates x and y) in the input image. Usually this value is

normalized. V_F and V_L are normalizing constants and M and W represent the constant synaptic weights. M and W are computed by using the inverse square rule [10]:

$$f(k,l) = 2 / \sqrt{k^2 + l^2}$$

Y stands for the output of the neuron and can only take a binary value of 0 or 1.
The linking effect can be modeled as follows:

$$U_{ij}[n] = F_{ij}[n] \cdot (1 - \beta \cdot L_{ij}[n]) \tag{3}$$

$U_{ij}[n]$ represents the internal activation of the neuron and β is the linking weight parameter.

The pulse generator determines the firing events in the model. In fact, the pulse generator is also responsible for the modeling of the refractory period. As the neuron produces a spike, its threshold is raised to prevent it from firing again in the near future (established by the parameter settings). The threshold is then decreased to allow the neuron to fire when its activation is increased.

$$Y_{ij}[n] = \begin{cases} 1, if\ U_{ij}[n] > \Theta_{ij}[n-1] \\ 0, otherwise \end{cases} \tag{4}$$

$$\Theta_{ij}[n] = e^{-\alpha_\Theta} \cdot \Theta_{ij}[n-1] + V_\Theta \cdot Y_{ij}[n] \tag{5}$$

In equations (4) and (5) $\Theta_{ij}[n]$ represents the dynamic threshold of the neuron while α_Θ and V_Θ are the time constant and the normalization constant respectively.

During the simulation, each iteration updates the internal activity and the output for every neuron in the network, based on the stimulus signal from the image and the previous state of the network. For each iteration the total number of firings (Equation 6) over the entire PCNN is computed and stored in a global array G.

$$G[n] = \sum_{ij} Y_{ij}[n], \quad \text{where n is the iteration } (n = 0 \ldots N-1) \quad (6)$$

The global array is then used at the next levels of the system (to compute the DFT of the global pulse signal).

- The Discrete Fourier Transform :
  We used the standard analysis equations to calculate the DFT

$$\operatorname{Re} X[k] = \sum_{i=0}^{N-1} G[i] \cos(2\pi k i / N), \quad k = 0 \ldots N/2 \quad (7)$$

$$\operatorname{Im} X[k] = -\sum_{i=0}^{N-1} G[i] \sin(2\pi k i / N), \quad k = 0 \ldots N/2 \quad (8)$$

Computing the DFT means basically correlating the input signal with each basis function.
The DFT yields two shorter signals to be analyzed. We used only the imaginary part of the DFT in further processing but a combination may be possible as well. Our choice had been motivated by experimental observations that show a relative stability of the real part over all the shapes used for testing. We also enhanced speed by using only the imaginary part in the higher levels.

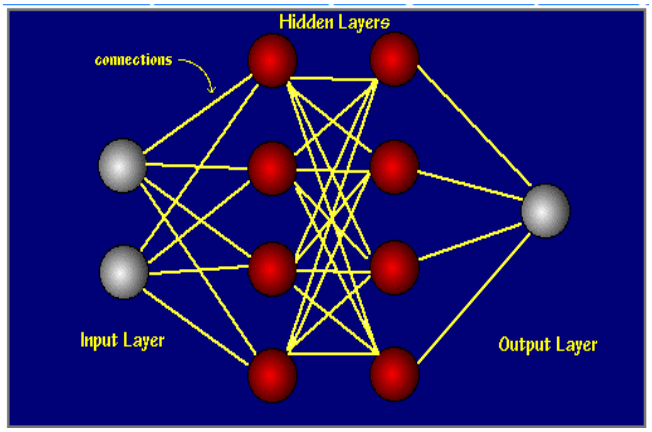
- The classifier:
  Our classifier is basically a multilayer perceptron (MLP). The neural architecture consists of one input layer, one hidden layer and one output neuron. The input layer contains a number of inputs equal to the samples in the imaginary part of the DFT signal (Im X in eq. Then, a hidden layer has an extension of about 10 to 20% of the input layer.
  Because of the specific tasks used to test the system, the output layer contained only one neuron (target detection). An output value of 1is equivalent to target detection whereas a value of 0 means no target detection.

  A standard backpropagation algorithm is used for supervised training.

D- Open CV: OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel research center in Nizhny

Novgorod (Russia), later supported by Willow Garage and now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.

Figure 4 Neural Network



E- Neural Network:

Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. Neural networks have advantages include: Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which send messages to each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning. For example, a neural network for handwriting recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

Tip : is to train the network more and more and select the appropriate factors like number of hidden neurons because this factor is very important in back propagation algorithm and learning rate too to avoid over fitting problem

What is a feed forward neural network? A feed forward neural network is a biologically inspired classification algorithm. It consist of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal, each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes. Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feedforward neural networks.
In the following figure we see an example of a 2-layered network with, from top to bottom: an output layer with 5 units, a hidden layer with 4 units, respectively. The network has 3 input units.

The 3 inputs are shown as circles and these do not belong to any layer of the network (although the inputs sometimes are considered as a virtual layer with layer number 0). Any layer that is not an output layer is a hidden layer. This network therefore has 1 hidden layer and 1 output layer. The figure also shows all the connections between the units in different layers. A layer only connects to the previous layer.

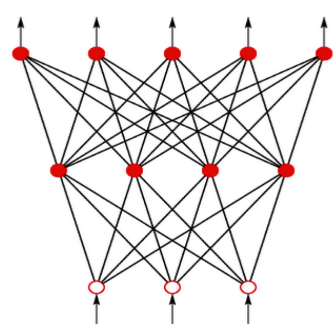


Figure 5 Neural Network

The operation of this network can be divided into two phases:

1. The learning phase

1.1- How does learning take place? The FFNet uses a supervised learning algorithm: besides the input pattern, the neural net also needs to know to what category the pattern belongs. Learning proceeds as follows: a pattern is presented at the inputs. The pattern will be transformed in its passage through the layers of the network until it reaches the output layer. The units in the output layer all belong to a different category. The outputs of the network as they are now are compared with the outputs as they ideally would have been if this pattern were correctly classified: in the latter case the unit with the correct category would have had the largest output value and the output values of the other output units would have been very small.

On the basis of this comparison all the connection weights are modified a little bit to guarantee that, the next time this same pattern is presented at the inputs, the value of the output unit that corresponds with the correct category is a little bit higher than it is now and that, at the same time, the output values of all the other incorrect outputs are a little bit lower than they are now. (The differences between the actual outputs and the idealized outputs are propagated back from the top layer to lower layers to be used at these layers to modify connection weights. This is why the term back propagation network is also often used to describe this type of neural network.

1.2- How long will the learning phase take? In general this question is hard to answer. It depends on the size of the neural network, the number of patterns to be learned, the number of epochs, the tolerance of the minimizer and the speed of your computer, how much computing time the learning phase may take.

1.3- The classification phase: In the classification phase the weights of the network are fixed. A pattern, presented at the inputs, will be transformed from layer to layer until it reaches the output layer. Now classification can occur by selecting the category associated with the output unit that has the largest output value.

2- Back propagation Neural Networks: Back propagation neural networks employ one of the most popular neural network learning algorithms, the Back propagation (BP) algorithm. It has been used successfully for wide variety of applications, such as speech or voice recognition, image pattern recognition, medical diagnosis, and automatic controls. Back propagation made a tremendous step forward from the single-layer perceptron network. With a more sophisticated learning rule, back propagation networks overcome the limitations that single-layer networks have. Back propagation is also the most suitable learning method for multilayer networks. Perhaps, the reason why the back propagation made the major turning point is because the learning rule has a solid mathematical foundation and it is practical.

Steps of the back propagation algorithm:-

We can now formulate the complete back propagation algorithm and prove by induction that it works in arbitrary feed-forward networks with differentiable activation functions at the nodes. We assume that we are dealing with a network with a single input and a single output unit.

Back propagation algorithm: - Consider a network with a single real input x and network function F. The derivative F(x) is computed in two phases: Feed-forward: the input x is fed into the network. The primitive functions at the nodes and their derivatives are evaluated at each node. The derivatives are stored.

Back propagation: the constant 1 is fed into the output unit and the network is run backwards. Incoming information to a node is added and the result is multiplied by the value stored in the left part of the unit. The result is transmitted to the left of the unit. The result collected at the input unit is the derivative of the network function with respect to x.

Back Propagation network learns by example. You give the algorithm examples of what you want the network to do and it changes the network’s weights so that, when training is finished, it will give you the required output for a particular input. Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks4 . As just mentioned, to train the network you need to give it examples of what you want – the output you want (called the Target) for a particular input as shown in

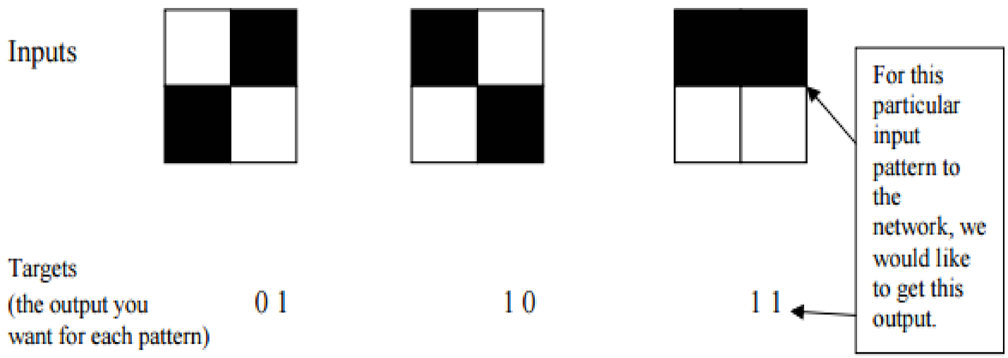


Figure 6 a Back Propagation training set.

So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in figure 7 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a Training Pair.

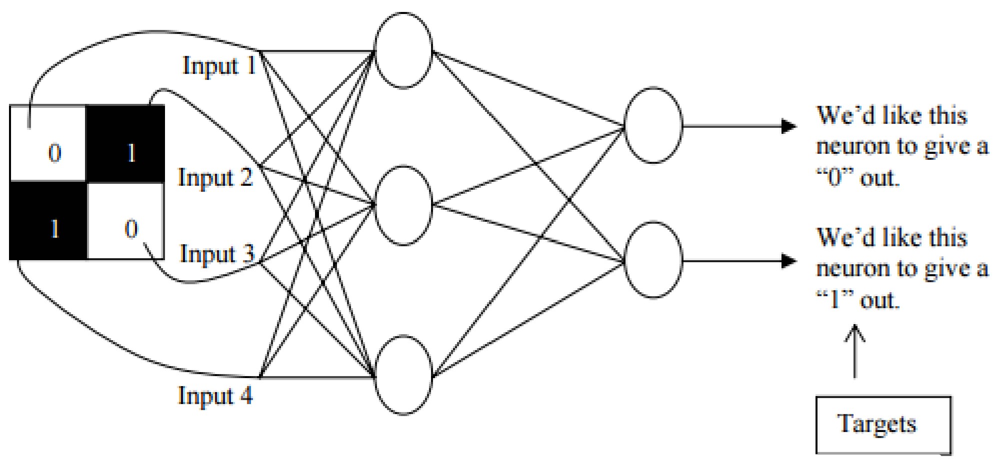


Figure 7, applying a training pair to a network.

Once the network is trained, it will provide the desired output for any of the input patterns. Let's now look at how the training works.

The network is first initialized by setting up all its weights to be small random numbers – say between –1 and +1. Next, the input pattern is applied and the output calculated (this is called the forward pass).

The calculation gives an output which is completely different to what you want (the Target), since all the weights are random. We then calculate the Error of each neuron, which is essentially:

Target - Actual Output (i.e. What you want – What you actually get). This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the reverse pass). The process is repeated again and again until the error is minimal.

Let's do an example with an actual network to see how the process works. We'll just look at one connection initially, between a neuron in the output layer and one in the hidden layer, figure 8.

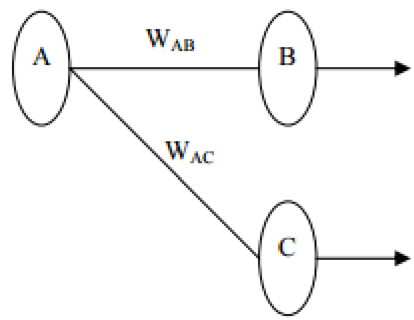


Figure 8, single connection learning in a Back Propagation network.

The connection we're interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron) and has the weight WAB. The diagram also shows another connection, between neuron A and C, but we'll return to that later. The algorithm works like this:

1- First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.
2- Next work out the error for neuron B. The error is What you want – What you actually get, in other words:

$$Error_B = Output_B\ (1-Output_B)(Target_B - Output_B)$$

The "Output(1-Output)" term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be (Target – Output).

3- Change the weight. Let W^{+}_{AB} be the new (trained) weight and WAB be the initial weight.

$$W^{+}_{AB} = W_{AB} + (Error_B \times Output_A)$$

Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way.

4- Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we Back Propagate them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.

$$Error_A = Output_A\ (1 - Output_A)(Error_B\ W_{AB} + Error_C\ W_{AC})$$

Again, the factor "Output (1 - Output)" is present because of the sigmoid squashing function.

5- Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

This may well have left some doubt in your mind about the operation, so let's clear that up by explicitly showing all the calculations for a full sized network with 2 inputs, 3 hidden layer neurons and 2 output neurons as shown in figure 9.

W^{+} represents the new, recalculated, weight, whereas W (without the superscript) represents the old weight.

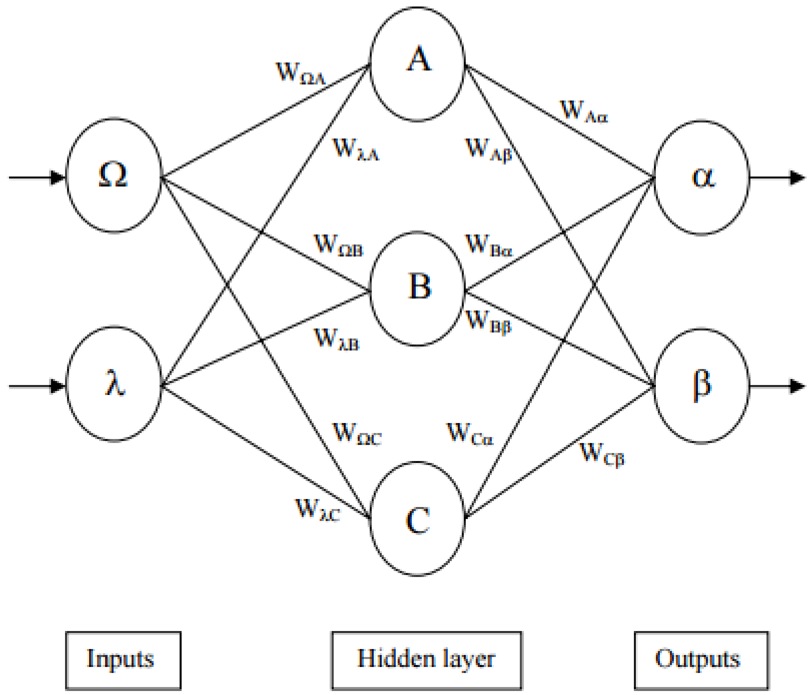


Figure 9, all the calculations for a reverse pass of Back Propagation.

1- Calculate errors of output neurons

$\delta\alpha = \text{out}\alpha\ (1 - \text{out}\alpha)\ (\text{Target}\alpha - \text{out}\alpha)$

$\delta\beta = \text{out}\beta\ (1 - \text{out}\beta)\ (\text{Target}\beta - \text{out}\beta)$

2- Change output layer weights

$W^{+}_{A\alpha} = W_{A\alpha} + \eta\delta_{\alpha}\ \text{out}_{A}$ $\quad$ $W^{+}_{A\beta} = W_{A\beta} + \eta\delta_{\beta}\ \text{out}_{A}$

$W^{+}_{B\alpha} = W_{B\alpha} + \eta\delta_{\alpha}\ \text{out}_{B}$ $\quad$ $W^{+}_{B\beta} = W_{B\beta} + \eta\delta_{\beta}\ \text{out}_{B}$

$W^{+}_{C\alpha} = W_{C\alpha} + \eta\delta_{\alpha}\ \text{out}_{C}$ $\quad$ $W^{+}_{C\beta} = W_{C\beta} + \eta\delta_{\beta}\ \text{out}_{C}$

3- Calculate (back-propagate) hidden layer errors

$\delta_{A} = \text{out}_{A}\ (1 - \text{out}_{A})\ (\delta_{\alpha}W_{A\alpha} + \delta_{\beta}W_{A\beta})$

$\delta_{B} = \text{out}_{B}\ (1 - \text{out}_{B})\ (\delta_{\alpha}W_{B\alpha} + \delta_{\beta}W_{B\beta})$

$\delta_{C} = \text{out}_{C}\ (1 - \text{out}_{C})\ (\delta\alpha W_{C\alpha} + \delta_{\beta}W_{C\beta})$

4- Change hidden layer weights

$W^{+}_{\lambda A} = W_{\lambda A} + \eta\delta_{A}\ \text{in}_{\lambda}$ $\quad$ $W^{+}_{\Omega A} = W^{+}_{\Omega A} + \eta\delta_{A}\ \text{in}_{\Omega}$

$W^{+}_{\lambda B} = W_{\lambda B} + \eta\delta_{B}\ \text{in}_{\lambda}$ $\quad$ $W^{+}_{\Omega B} = W^{+}_{\Omega B} + \eta\delta_{B}\ \text{in}_{\Omega}$

$W^{+}_{\lambda C} = W_{\lambda C} + \eta\delta_{C}\ \text{in}_{\lambda}$ $\quad$ $W^{+}_{\Omega C} = W^{+}_{\Omega C} + \eta\delta_{C}\ \text{in}_{\Omega}$

The constant η (called the learning rate, and nominally equal to one) is put in to speed up or slow down the learning if required. To illustrate this let’s do a worked Example.

Consider the simple network below:

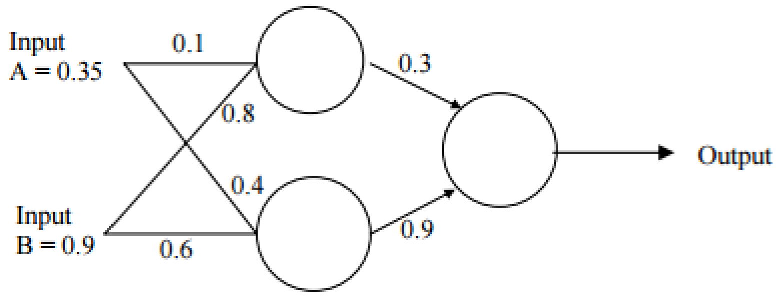


Figure 10 example 1

Assume that the neurons have a Sigmoid activation function and

(i) Perform a forward pass on the network.
(ii) Perform a reverse pass (training) once (target = 0.5).
(iii) Perform a further forward pass and comment on the result.

Answer:

(i) Input to top neuron = (0.35x0.1)+(0.9x0.8)=0.755.
Out = 0.68. Input to bottom neuron = (0.9x0.6)+(0.35x0.4) = 0.68.
Out = 0.6637. Input to final neuron = (0.3x0.68)+(0.9x0.6637) = 0.80133. Out = 0.69.

(ii) Output error δ=(t-o)(1-o)o = (0.5-0.69)(1-0.69)0.69 = -0.0406.

New weights for output layer
$w1^+ = w1^+(\delta \times \text{input}) = 0.3 + (-0.0406 \times 0.68) = 0.272392$.
$w2^+ = w2+(\delta \times \text{input}) = 0.9 + (-0.0406 \times 0.6637) = 0.87305$.

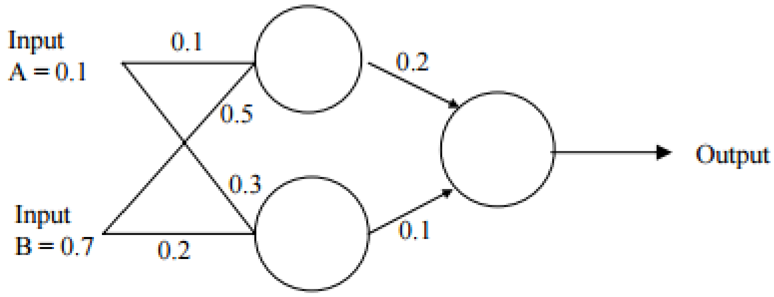
Errors for hidden layers:

δ1 = δ x w1 = -0.0406 x 0.272392 x (1-o)o = -2.406x10-3
δ2= δ x w2 = -0.0406 x 0.87305 x (1-o)o = -7.916x10-3

New hidden layer weights:

$w3^+$ =0.1 + (-2.406 x 10-3 x 0.35) = 0.09916.
$w4^+$ = 0.8 + (-2.406 x 10-3 x 0.9) = 0.7978.
$w5^+$ = 0.4 + (-7.916 x 10-3 x 0.35) = 0.3972.
$w6^+$ = 0.6 + (-7.916 x 10-3 x 0.9) = 0.5928.

(iii) Old error was -0.19. New error is -0.18205. Therefore error has reduced.
Try a training pass on the following example. Target = 1, Learning rate = 1:



Running the algorithm:-

Now that we've seen the algorithm in detail, let's look at how it's run with a large data set. Suppose we wanted to teach a network to recognize the first four letters of the alphabet on a 5x7 grid, as shown in figure11.

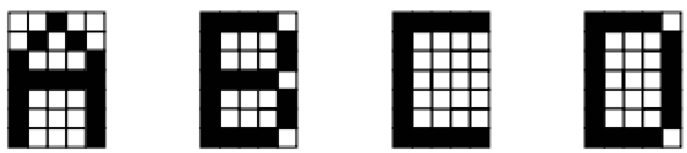


Figure 11 the first four letters of the alphabet.

The correct way to train the network is to apply the first letter and change ALL the weights in the network ONCE. Next apply the second letter and do the same, then the third and so on. Once you have done all four letters, return to the first one again and repeat the process until the error becomes small (which means that it is recognising all the letters).

Figure 12 summarizes how the algorithm should work.

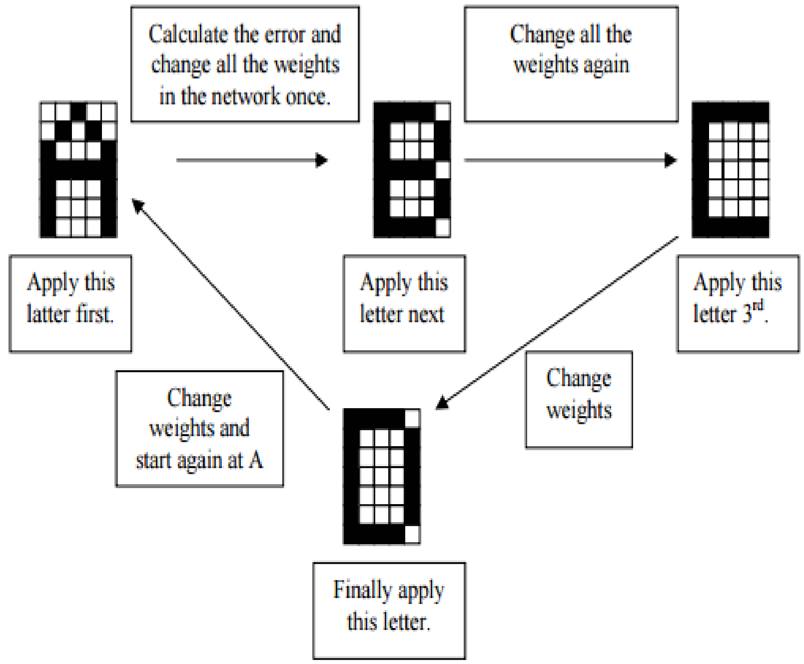


Figure 12, the correct running of the algorithm.

One of the most common mistakes for beginners is to apply the first letter to the network, run the algorithm and then repeat it until the error reduces, then apply the second letter and do the same. If you did this, the network would learn to recognize the first letter, then forget it and learn the second letter, etc and you'd only end up with the last letter the network learned.

Stopping training:- When do we stop the training? We could stop it once the network can recognize all the letters successfully, but in practice it is usual to let the error fall to a lower value first. This ensures that the letters are all being well recognized. You can evaluate the total error of the network by adding up all the errors for each individual neuron and then for each pattern in turn to give you a total error as shown in figure13.

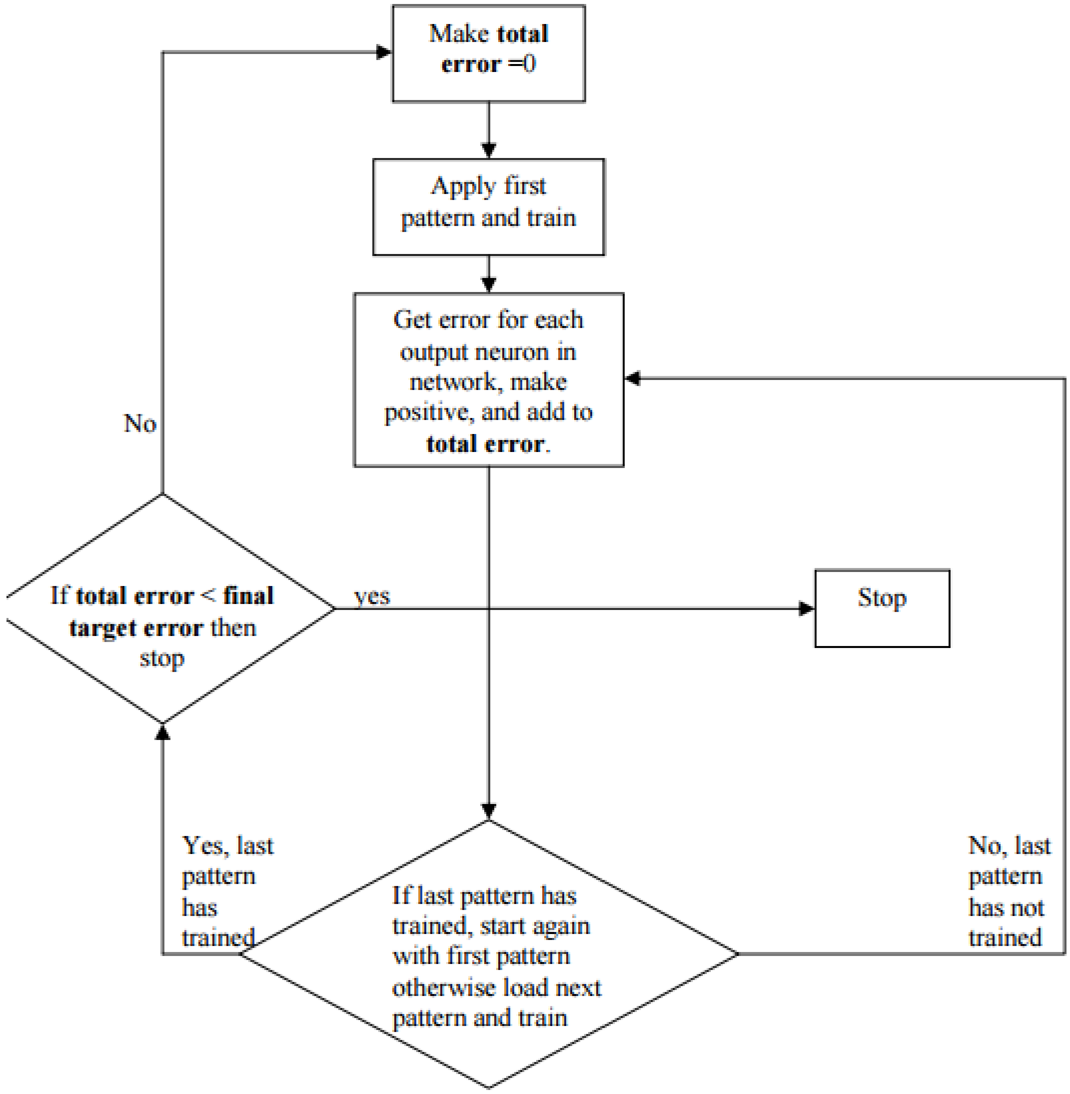


Figure 13, total error for network.

In other words, the network keeps training all the patterns repeatedly until the total error falls to some pre-determined low target value and then it stops. Note that when calculating the final error used to stop the network (which is the sum of all the individual neuron errors for each pattern) you need to make all errors positive so that they add up and do not subtract (an error of -0.5 is just as bad as an error of +0.5).

Once the network has been trained, it should be able to recognise not just the perfect patterns, but also corrupted or noisy versions as was explained in section 2.1. In fact if we deliberately add some noisy versions of the patterns into the training set as we train the network (say one in five), we can improve the network's performance in this respect. The training may also benefit from applying the patterns in a random order to the network.
There is a better way of working out when to stop network training - which is to use a Validation Set. This stops the network overtraining (becoming too accurate, which can lessen its performance). It does this by having a second set of patterns which are Make total error =0 Apply first pattern and train Get error for each output neuron in network, make positive, and add to total error. If total error < final target error then stop If last pattern has trained, start again with first pattern otherwise load next pattern and train yes Stop No No, last pattern has not trained Yes, last pattern has trained 24 noisy versions of the training set (but aren't used for training themselves). Each time after the network has trained; this set (called the Validation Set) is used to calculate an error. When the error becomes low the network stops. Figure 14 shows the idea.

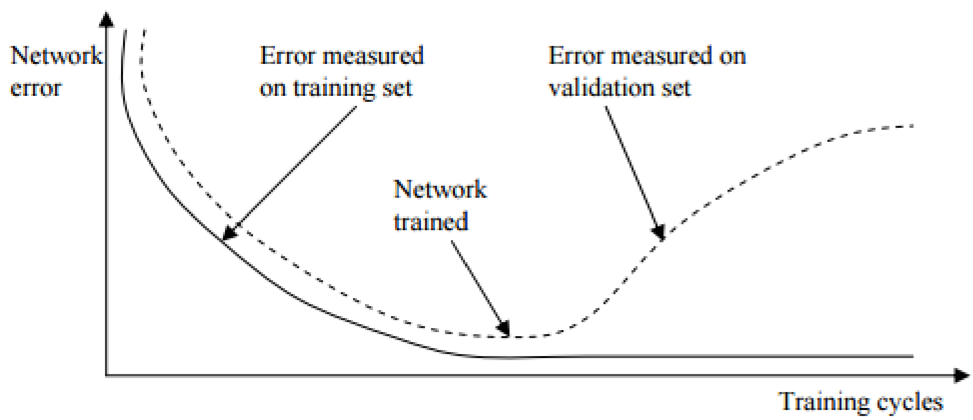


Figure 14 use of validation sets

When the network has fully trained, the Validation Set error reaches a minimum. When the network is overtraining (becoming too accurate) the validation set error starts rising. If the network over trains, it won't be able to handle noisy data so well.

Problems with Back propagation:-

Back propagation has some problems associated with it. Perhaps the best known is called “Local Minima”. This occurs because the algorithm always changes the weights in such a way as to cause the error to fall. But the error might briefly have to rise as part of a more general fall, as shown in figure 15. If this is the case, the algorithms will “gets stuck” (because it can’t go uphill) and the error will not decrease further.

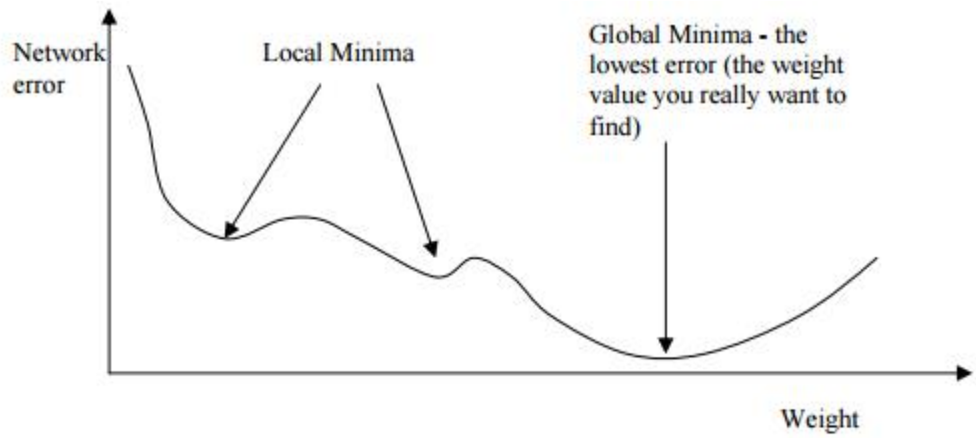


Figure 15, local minima.

5- There are several solutions to this problem. One is very simple and that is to reset the weights to different random numbers and try training again (this can also solve several other problems). Another solution is to add “momentum” to the weight change. This means that the weight change this iteration depends not just on the current error, but also on previous changes. For example:

W+ = W + Current change + (Change on previous iteration * constant)

Constant is < 1.

There are other lesser known problems with Back propagation as well. These tend to manifest themselves as the network gets larger, but many can be overcome by reinitializing the weights to different starting values. It’s also worth noting that many variations on the standard Back propagation Algorithm have been developed over the years to overcome.

3- Problem Definition: Has anyone ever thought that one day he can speak with Mute-deaf people else without difficulty in understanding?
Has anyone ever thought that one day he can talk with Mute-deaf people without troubles?
The dealing with Mute-deaf people is an important part in our life. It is known that they used sign language by hand gestures, so their hand gestures can be used to know what they want to tell. What about hand gestures reading, is it really that thoughts can be read. No!!
So can anyone read something other say sign language?
It's a big challenge these days to make life simpler

4- Problem objective**:** Mute-deaf gestures recognition is a project which helps Mute-deaf people to express what they want, especially when talking with ordinary people and also facilitates communication between the different categories of society and to reduce the problems of their lives and even be available everywhere and easy to carry and easy to deal with the program.

5- Related work: In hand gesture and recognition system there are two phases hand detection, hand gesture recognition.

A- Hand Detection: For hand detection image is taken from camera. System takes a five image as input. These image processed in system. The operators used for image processing must be kept low time consuming in order to obtain the fast processing rate needed to achieve real time speed.

There are many approaches for hand detection. The simple way to detect hand is capture image and find for skin color region in the image but skin color region detection is difficult because it can also detect background color and other body parts from image. The camera is used to track the hand movements. So we use the skin color detection algorithm for skin color detection.

B- Hand Gesture Recognition: In hand gesture recognition two phases are important, firstly features detection which relates with the extraction of useful features from input video or input image, Secondly, relates with calculation of parameters estimation model from the extracted features. Hand gesture can be localized by detecting the hand gesture from the image and segmenting hand from the background which is the unwanted other objects. Skin color provides an effective and efficient method for hand localization.

Segmentation based skin color method applied for hand locating. Recognition process affected with the proper selection of gesture parameters of features and thus the accuracy of the classification. For example edge detection and counter not suitable for gesture recognition since it might lead to misclassification. Edge detection algorithm is applied on the image captured by camera. These algorithms detect which gesture is selected. Once you select hand gesture system retrieves the information from internet with respect to hand gesture.

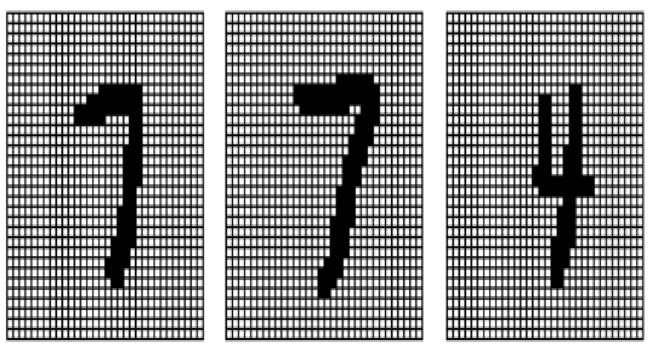
6- The Euclidean Distance of Images: We present a new Euclidean distance for images, which we call Image Euclidean Distance (IMED). Unlike the traditional Euclidean distance, IMED takes into account the spatial relationships of pixels. Therefore it is robust to small perturbation of images. We argue that IMED is the only intuitively rea-sonable Euclidean distance for images. IMED is then applied to image recognition. The key advantage of this distance measure is that it can be embedded in most image classification techniques such as SVM, LDA and PCA. The embedding is rather efficient by involving a transformation referred to as Standardizing Transform (ST). We show that ST is a transform domain smoothing. Using the Face Recognition Technology (FERET) database and two state-of-the-art face identification algorithms, we demonstrate a consistent performance improvement of the algorithms embedded with the new metric over their original versions.

Introduction: A central problem in image recognition and computer vision is determining the distance between im-ages. Considerable efforts have been made to define image distances that provide intuitively reasonable results [4, 16, 1, 10]. Among others, two representative measures are the tangent distance [16] and the generalized Hausdorff distance [4]. Tangent distance is locally invariant with respect to some chosen transformations, and has been widely used in handwritten digit recognition. The generalized Hausdorff distance is not only robust to noise but also allows portions of one image to be compared with another,and has become a standard tool for comparing shapes. However, from the image recognition point of view, many existing image distances suffer from some of the following disadvantages:

1) It is difficult to combine the metric with those powerful image recognition techniques such as SVM,LDA, PCA, etc
2) The computation of the measure is complicated.
3) The distance does not obey the triangle inequality, so sometimes two highly dissimilar images can be both similar to an unknown object.

❖ Among all the image metrics, Euclidean distance is the most commonly used due to its simplicity. Let x; y be two M by N images, $x = (x^1; x^2; \ldots; x^{MN})$, $y = (y^1; y^2; \ldots;$ yMN), where x^{kN+l}; $y^{kN+l}$are the gray levels at location (k; l). The Euclidean distance $d_E(x; y)$ is given by

$$d^2_E(x; y) = \sum_{K=1}^{MN} (X^k - Y^k)^2$$


(a) (b) (c)

However, this distance measure suffers from a high sensitivity even to small deformation. Fig. shows three 32 by 32 images. A reasonable image metric should present smaller distance between (a), (b) than that of (a), (c). But the Euclidean distance gives counter intuitive result. For simplicity, let the gray levels be one at the black pixels and zero elsewhere. Computing the Euclidean distances yields $d_E(a; b) = 54$ and $d_E(a; c) = 49$. The pair with more similarity has a larger Euclidean distance! This phenomenon is caused by the fact that the Euclidean distance defined in (1) does not take into account that x; y are images, $x^k; y^K$ are gray levels on pixels.

For images, there are spatial relationships between pixels. The traditional euclidean distance is only a summation of the pixel-wise intensity differences, and consequently small deformation may result in a large Euclidean distance. This paper proposes a new Euclidean distance, which we call Image Euclidean Distance (IMED). Unlike the traditional one, IMED takes into consideration the spatial relationships of pixels. Based on three properties that (arguably) any intuitively reasonable image metric should satisfy, we show that IMED is the only Euclidean distance possessing these properties. IMED is then applied to image recognition. The key advantages of this metric are:

1) Relative insensitivity to small perturbation (deformation);
2) Simplicity of computation;
3) It can be efficiently embedded in most of the powerful image recognition techniques.

In order to simplify the computation of IMED, we introduce an image transformation referred to as Standardizing Transform (ST). We show that ST is a transform domain smoothing. This result directly relates image Euclidean distance to smoothing. It indicates that smoothing noiseless images can still increase the recognition rate.

Chapter 2

Public knowledge about project Work

One of the earliest written records of a sign language is from the fifth century BC, in Plato's Cratylus, where Socrates says: "If we hadn't a voice or a tongue, and wanted to express things to one another, wouldn't we try to make signs by moving our hands, head, and the rest of our body, just as dumb people do at present?"

In 2014, according to the World Federation of the Deaf (WFD), over 70 millions deaf & mute people are use sign language as their first language. In Bangladesh their number is 2.6 millions. So their percentage is 1.733%. Many of deaf & mute people are now holding different position in different place of work. Some of them are student, technician, labor worker etc as well as teacher, officer, manager.

In recent years there has been a lot of research on hand gesture recognition. Several techniques have been reported on gesture recognition which includes skin segmentation using color pixel classification , region growing by exemplar-based hand segmentation under complex background , Parametric Hidden Markov models for gesture recognition , statistical database comparison

method , accelerometer-based gesture recognition system, orientation histograms for gesture recognition , Finger Detection for Sign Language Recognition etc. Most of the gesture recognition systems use special devices like hand glove. The gloves get connected to the computers using a lot of cables. So these devices are cumbersome and expensive. In order to overcome these difficulties, alternatively vision-based approaches involving camera and image processing for recognizing gestures are being explored. There has been previous work on use of features for finger detection. Some common features extracted include hand silhouettes, contours, key points distributed along fingertips & joints. There has also been reported work where finger detection has been accomplished via color segmentation, real-time hidden

Markov model-based systems, computational model of intelligibility for ASL (CIM-ASL), enhanced level building (eLB) algorithm and contour extraction. But color based segmentation techniques requires fine-tuning every time the system switches to a new user as the color complexion varies from person to person. With the limitations posed by the schemes discussed above there is scope to devise a more efficient and robust technique.

There are few mobile applications for Deaf and dumb like Deaf and Dumb through 3G applications. These techniques only enable communication between deaf and dumb through sign language using mobile phones .The mobile application which helps to make recognition of Sign language. Mobile-based Deaf and Dumb Interaction System project which is proposed mobile application that enables the needs of 'deaf and dumb developing a voice-activated mobile which would convert their sign language into messages that may be read by other users, this message can also converted to a voice.

Sign Language: Just like spoken languages, sign language is built upon certain rules of grammar, and can also be varied in terms of dialects. But unlike spoken languages, dialects in sign language are not as comprehensible between people using related dialects. For example, British sign language is almost unintelligible for users of American Sign Language. On the other hand, American Sign Language has a similarity of 60% to modern French sign language, which demonstrates the difference in relationship and affinity between different sign languages and the equivalent spoken languages.

Technology: To bridge the communication gap between deaf-mute people and the people in their surroundings, there are some current methods put in use. Below is presented some of the technologies that are used to carry out these methods together with technologies with the potential of being used in future communication solutions.

1- Augmentative and Alternative Communication (AAC) is an umbrella term for all types of communication enhancing methods used by people (except oral communication) to express themselves to others. This includes body language and facial expressions. AAC is further categorized in unaided communication systems, where the user’s body is relied on for communication, and Aided communication systems, where devices ranging from pencils to computers that produce voice output are utilized in addition to the user's body.

Different kinds of relay services are utilized to help deaf-mute people communicate with hearing people. Video Remote Interpreting (VRI) and Video Relay Service (VRS) are two similar services, where VRI is used for communication between people at the same location, and

VRS is used to interpret messages between people at different locations. Both services rely on Internet and video communications technology, since the interpreters are never located at the same location as the people using the service. Text Relay Service (TRS) is similar to VRS, with the difference that the output to the person with hearing impairments is text instead of video. Keyboards or special assistive devices are used to send text messages to standard telephones via the telephone line. IP Relay Services are web-based, similar to chats, and do not rely on telephones at all. Thus, callers have to manually supply the operators with their location information during situation such as making emergency calls.

2- The Leap Motion is a sensor device that monitors hand and finger motion in order to use these as input to a computer, e.g. to control different kinds of interfaces. The device was initially created to overcome the cumbersome process of 3D modeling with mouse and keyboard as input devices, but is currently used within a large area of use, such as controlling computer games, web browsers and virtual musical instruments. Recent attempts have also been made to use the Leap Motion as a gesture-based sign translator for online chat applications.

3- The Kinect is also a motion sensing device, but unlike Leap Motion, Kinect monitors full-body motion. Usually the sensor device is positioned on top of a monitor, which displays the interface of whatever is being controlled with the sensor device. Recent efforts have been made to build a sign language to text/text to sign language translator using the Kinect for sign language input.

4- Machine translation is a subfield of computational linguistics that investigates how to translate text or speech from one natural language to another. Great efforts are currently being put into making results produced through machine translation more accurate. Corpus linguistics and statistical techniques are utilized to be able to recognize whole phrases of text or speech instead of the single words by themselves.

Chapter 3

Analysis and Design

1. Analysis

1.1 Determine problem: Communications between deaf-mute and a normal person have always been a difficult task. The deaf- mute have their own manual-visual language known assign language. This languages do not have a common origin and hence difficult to interpret. Deaf-Mute communication interpreter is a device that translates the hand gestures to voice. A hand gesture inside a sign language, is a particular movement of the hands with a specific shape to consider what he want to speak, we decide to solve this problem in this project.

1.2 Scope

1.2.1 In this product software:-

- Translate sign language to voice**.**
- Run on desktop version and mobile version
- When user Capture number of pictures , he should move hands with normal move speed
- The product will be available as mobile application in android

1.2.2 Not In this product software:-

- Standing far away from the camera
- Hand move quickly
- There is no translated at the same time take the picture

1.3 Solution : we tried to make adapter between deaf and mute person and other people to facilitate dealing with them so, we did a software that available as mobile application (Android) and desktop application that translate sign language to voice using Open CV, Neural Network (machine learning) and Pulse-Coupled Neural Networks

1.4 Requirements specification :

- **Overview**

- The focus of this Software is the development of application called Deaf and Mute helper
- When the application run it will open camera using Open CV
- Application will take a video to deaf person who speak using sign language
- Open CV will run and divide video into number of pictures
- Each picture will enter to PCNN code and return signature
- Saving all signatures for each word
- Run feed forward Neural Network for each Signature and return word meaning
- Application will return sound showing the meaning of words
- Terminate application or renter new video
- Application should make back propagation to learn new words
- Application should have easy and good interface
- Project should finished in 1/5/2015with its documentation

1.4.1 Functional Requirements

- Application should take a video of deaf and mute
- Entering this video into PCNN and Results out
- Entering this Results to Feed forward neural network
- Comparing result out with Results proper
- Running Back-propagation neural network for machine learning
- Application produce a voice that represent deaf person signs

1.4.2 External Interface Requirements

- **User Interface**

❖ The system has two buttons for the user to choose start button to take a video to deaf person or exit button to terminate application

- **Hardware Interfaces**

- ❖ It must be camera on desk top for Desktop version and for mobile version it must have front camera

- **Software Interfaces**

- ❖ The user will use camera‘s phone to capture image of a particular movement of
  the hands with a specific shape to consider what he want to speak using OPENCV API
- ❖ We will use J FREE Chart for draw line charts to consider the signature of the image and other things

1.4.3 Non-Functional Requirements

- **Usability**

- ❖ Application rules must be clear for the user to understand what the application need
- ❖ GUI: simple and good GUI to help user and every one can use application
  - We aim to make good and easy application for all people
  - Availability of software as mobile application
  - Achieve best result for predicting the meaning of sign language
  - Achieve usable and good interface
  - Help deaf and mute people

- **Reliability**

- Application should be available with no errors ~= 85% so is best expectation for deaf person signs

- **Performance**

- ❖ The application must be real time application so it’s need to have high performance

- **Supportability**

- ❖ The product supported by Android or IOS or both platform and pcs.

1.4.4 Organizing the Specific Requirement

- **System Mode**

  Depend on mode of operation that is consist of :-

1- Classes related to OPENCV of capture image or video

2- Classes related to PCNN to get image segmentation
3- Classes related to Pack propagation and feed forward
4- Classes related to draw line diagram of comparing saved result before with result now

- **User Class**

❖ User uses function in this product :-
1- Open application
2- Capture image or video
3- Exit the application

- **Objects**

❖ Every classes have set of attribute , function :-
1- In Class PCNN
1- PCNN() constructor
2- Clear ()
3- Save()
4- Other……
2- In Class line_chart2
1- Line_chart2() constructor
2- CreateDataset()
3- Readvector()

3- In pack propagation Class Matrix
1- SetMatrixDim()
2- Matrix() constructor
3- operator* : two matrix multiplication
4- other …….

4- in feed forward Class Feed Forward Algorithm
1- scale()
2- descale()
3- run()

1.4.5 Use Case Model

- **Use Case Model**

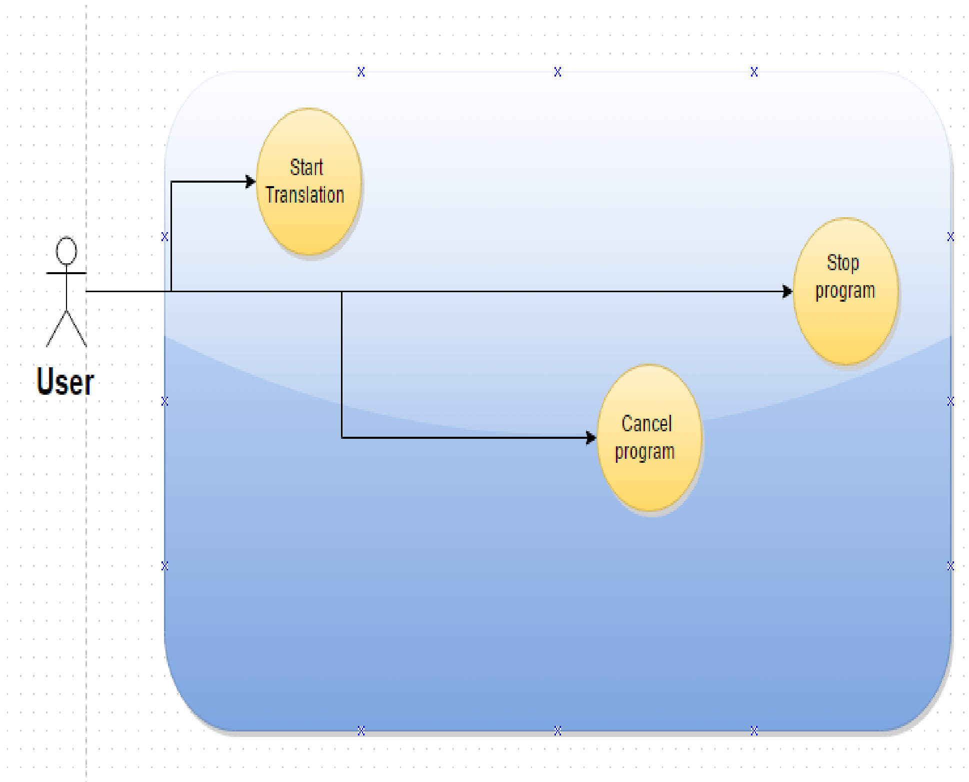


Figure 16 Use case

Activity Diagram

- **Use Case 1**

Use Case ID:	1
Use Case Name:	Start Translation
Actor:	deaf and mute person or any user want to learn this language
Description:	When the actor click on Start Translation of desktop version or mobile version
Preconditions:	Camera opened
post conditions:	Telephone should be working Or Desktop version should be working
Normal Flow:	1-open Desktop version or mobile version 2- open camera on desktop for Desktop version for mobile version it must have front camera
Frequency of Use:	When the actor want to speak with ordinary people

- **Use Case 2**

Use Case ID:	2
Use Case Name:	Stop
Actor:	deaf and mute person or any user want to learn this language
Description:	When the actor click on Stop record video of desktop version or mobile version
Preconditions:	Stop Capture record video
Post conditions:	Camera should be opened Camera capture record video
Normal Flow:	1- open Desktop version or mobile version 2- open camera on desktop for Desktop version for mobile version it must have front camera 3-Stop record video
Frequency of Use:	When the actor want to Stop Capture record video

- **Use Case 3**

Use Case ID:	3
Use Case Name:	Cancel program
Actor:	deaf and mute person or any user want to learn this language
Description:	When the actor click on Cancel program
Preconditions:	Cancel program
post conditions:	Start Translation
Normal Flow:	1- open Desktop version or mobile version 2- open camera on desktop for Desktop version for mobile version it must have front camera 3-Stop record video gramor cancel pro
Frequency of Use:	When the actor want to cancel program

2. Design

2.1 Project Architecture:

The environment of this proposed system is built in home environment and provides daily information. A camera is the major hardware component of system. Camera is used to captures the hand movements. These images are sending to PC for image processing. Image is processed in PC i.e. hand is detected and gesture is recognizedfrom the image. With respect to hand gesture information is retrieved from the internet. These information displays on the PC.If hand gesture does not identified then camera captures the images until gesture recognizes.

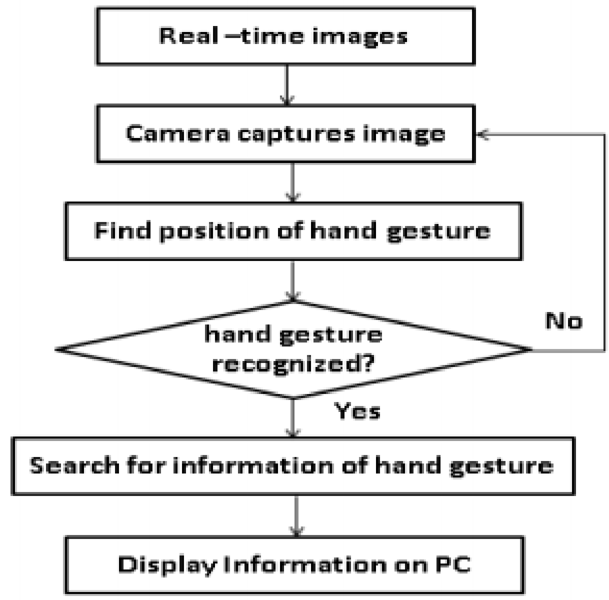


Figure 17 proposed system

2.2 Working:

In this concept the information is retrieved from date file using by hand gesture. An input is taken from camera. Camera captures the images and captured images are taken to processing. Processing means to identify hand gesture. For identifying hand gesture various algorithms are used. In this first we use skin color detection algorithm for detecting hand but it difficult to detect hand because sometimes this algorithm detects background color or background in the image as a skin color. Another algorithm is to detect edge from the image to identify gesture. Counting algorithm is used in gesture recognition process. If gesture is recognized then system goes for further processing i.e. collecting information from internet displays them on user's pc. If hand gesture does not recognize then system goes for capturing images until system recognize gesture.

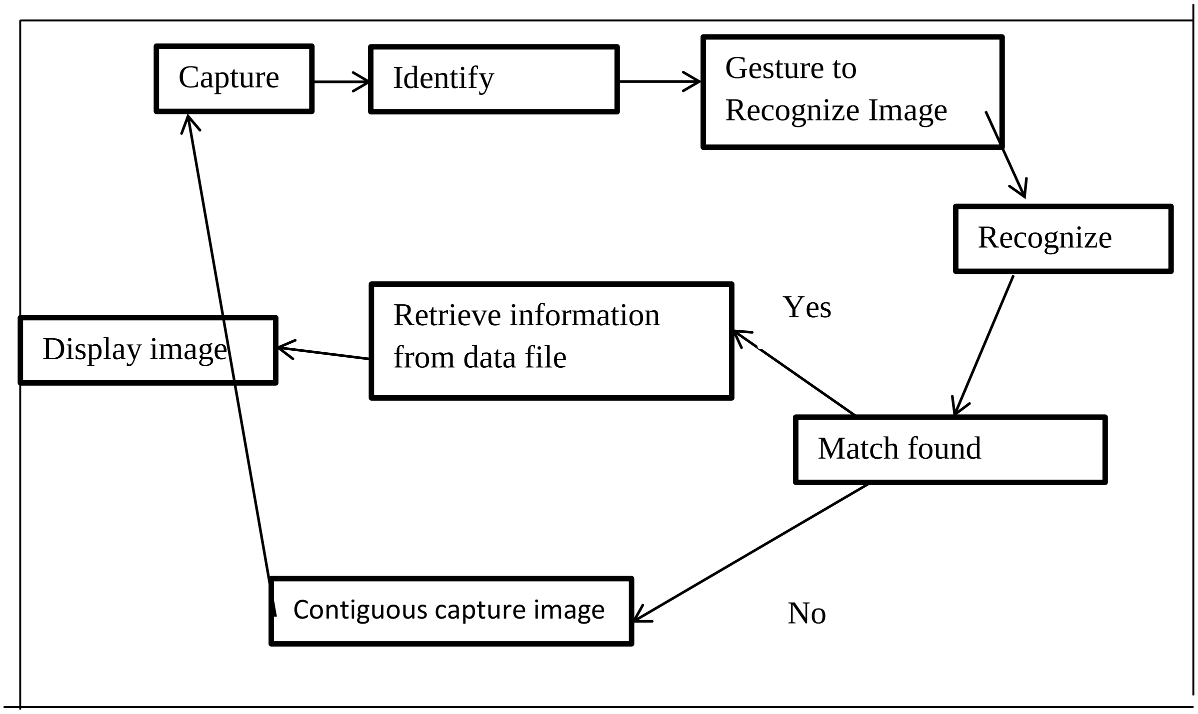


Figure 18 working system

- At first , User is opening the camera of the device he/ she uses it, application will running and check if there is a camera in device or no, using Open CV application will take a video and then divide it into number of pictures that express for meaning of sign of deaf person
- We will do preprocessing on the pictures like Scaling and graying all pictures to reach
- Entering each picture to PCNN and produce signature for each picture then, save it to file
- For each new pictures we

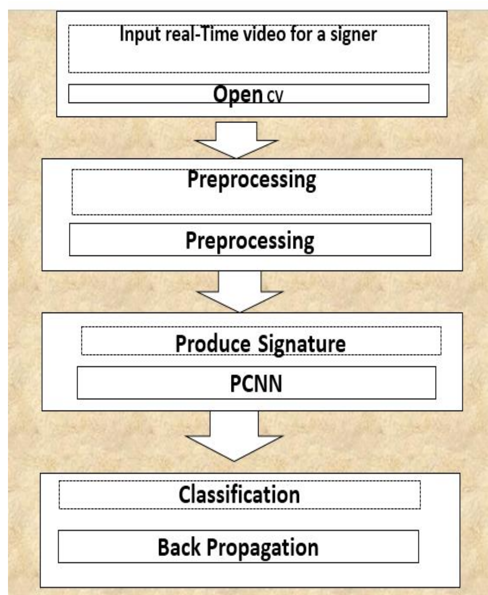


Figure 19 Project Architecture

- will run back propagation code in these signature and save weights for each word with it is meaning in data file
- Running feed forward code on signature that resulted from PCNN and data file so, we will knowing the meaning of sign

2.3 Class Diagram

2.3.1 UML for PCNN

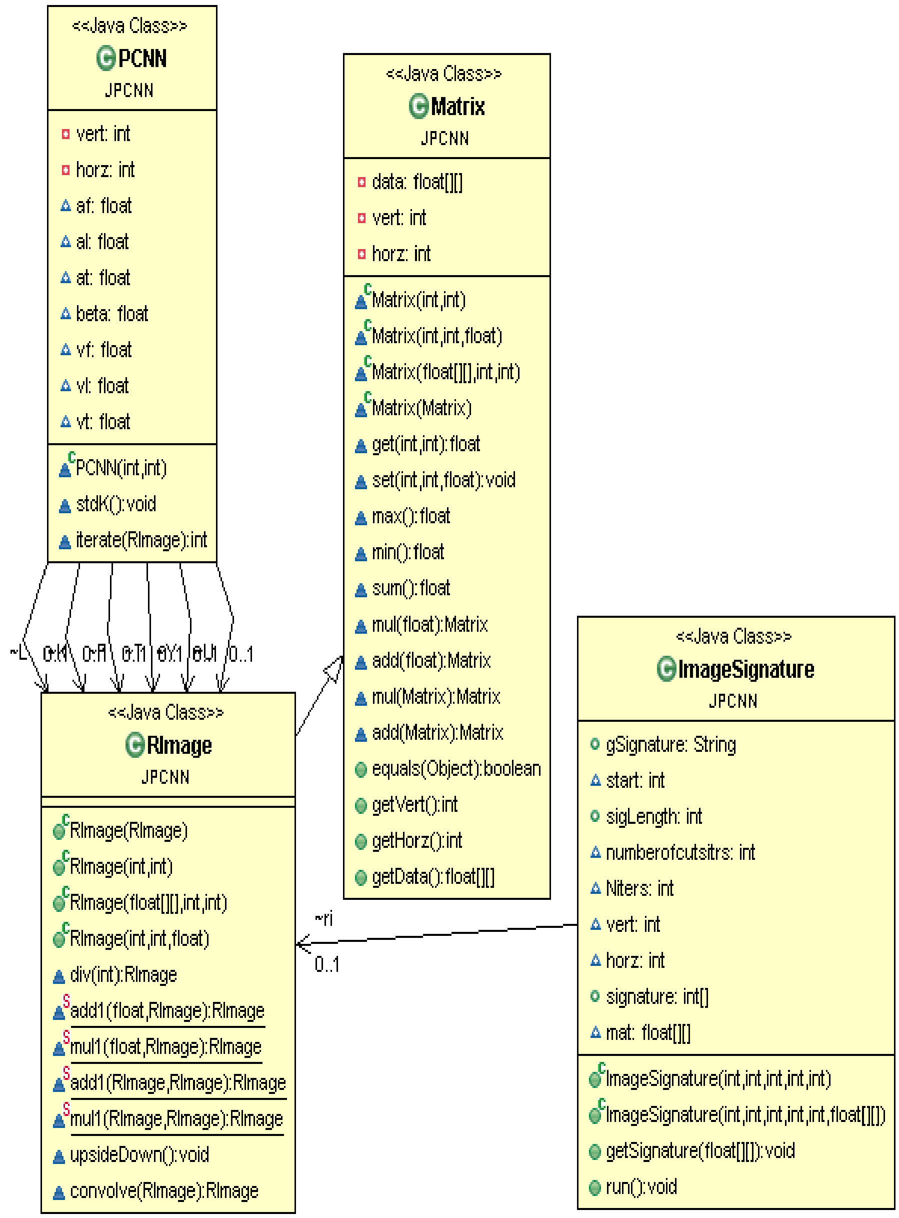
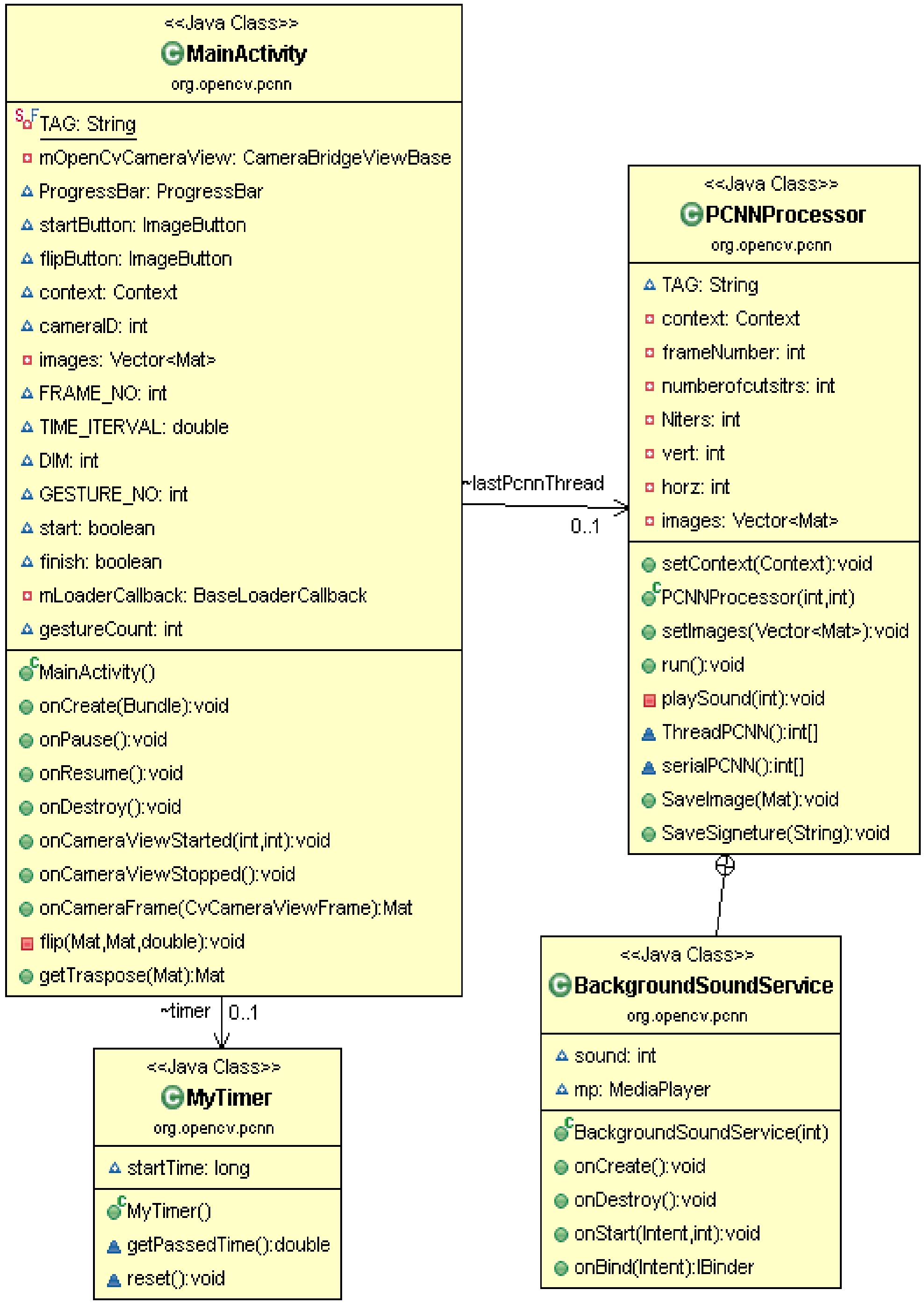


Figure 20 UML OF PCNN

2.3.2 UML for Open CV



2.3.3 UML for Back propagation

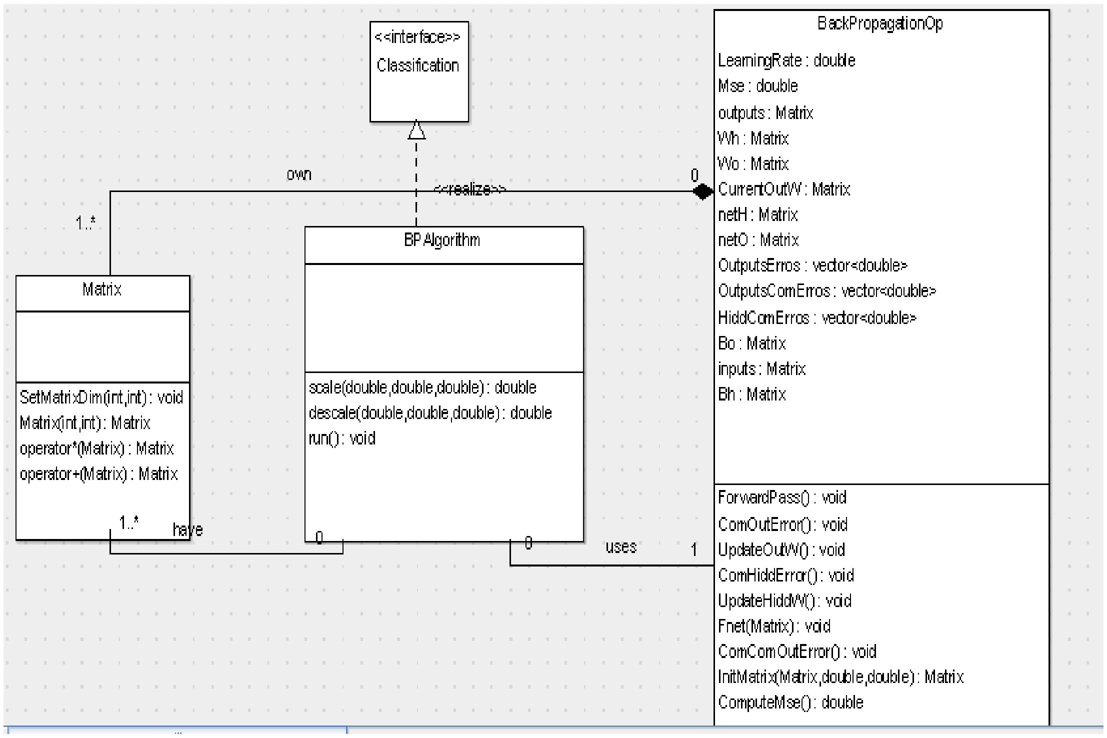


Figure 21 UML for Back propagation

2.3.4 UML for Feed Forward

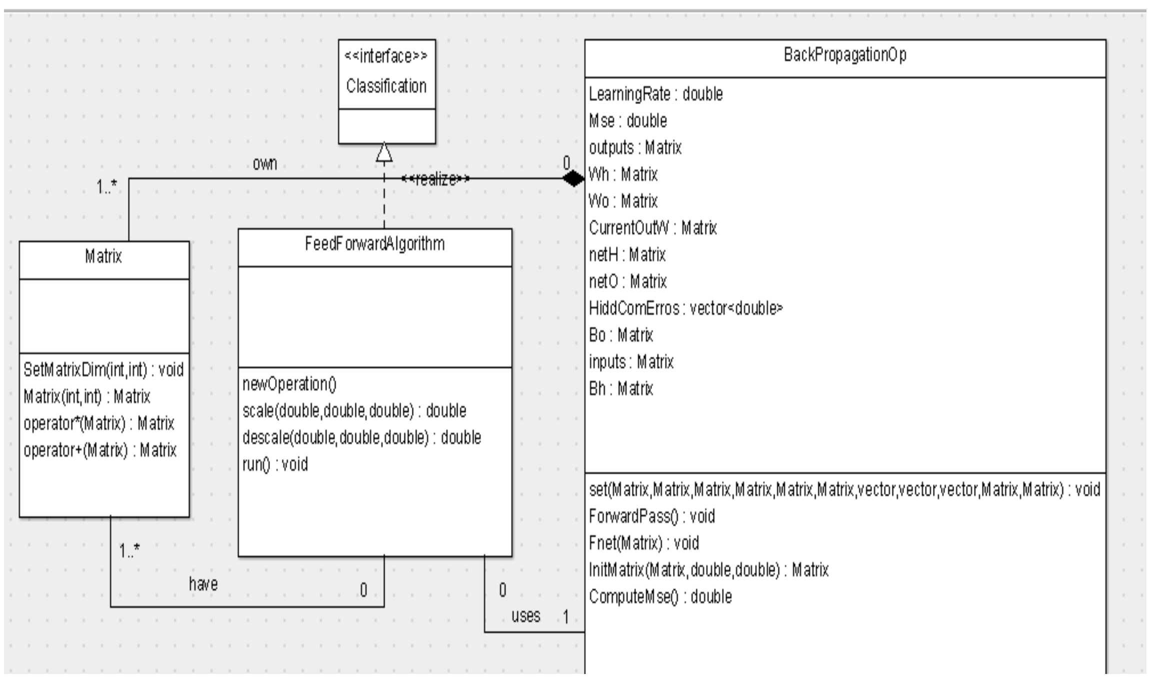


Figure 22 UML for Feed Forward

3. Proposed methodology consists of three phases, a training phase, a testing phase and recognition phase as figure 12. In the training phase, we used neural network (feed forward and Euclidean distance image comparison). In testing phase make combination feature vector (segmentation of gray scale image and signature by using PCNN and other processing that already explained in front of this chapter). In recognition phase uses NN (Neural networks and Euclidean distance image comparison) to check matching. Finally, after recognition of input image, their meaning is displayed on screen. Minimum Euclidean distance is calculated between test and train image and gesture is recognized. Recognized gesture is converted into text, voice format and also respective features will be display on GUI screen.

   Fig.6. shows a snapshot of application working and detecting two different hand gestures for sign H and Sign O in real time.

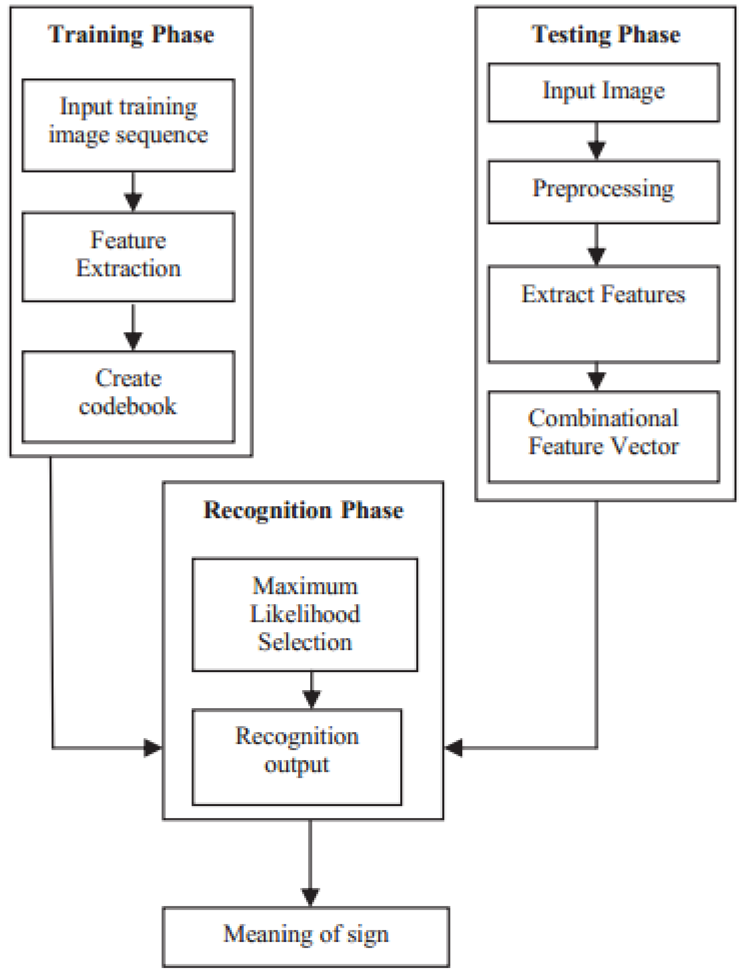


Figure 12 proposed system

4. Preprocessing

Preprocessing is applied to image before extracting feature from hand images. Steps of Preprocessing:

1- Segmentation of gray scale image as figure 13.

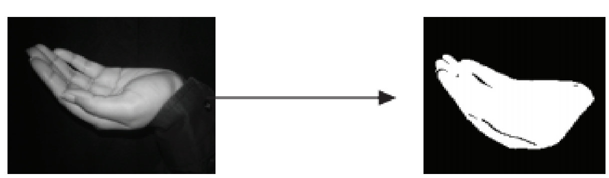


Figure 13 Segmentation of gray scale image

5. **Risks of this product:-**

1- Risk Analysis:

- As with any project, this undertaking is not entirely risk-free. Three major risk categories have been identified: Time, resource, and functionality.

1.1- Time Risks: -

As course requirements specify that the project must be completed within two academic semester , any Extensions are not possible. This introduces the risk that the system may not be completed with the full Functionality (not all words covered), this our time risk so that you must determine the scope of your project and the deliverables that you introduce

1.2- Resource Risks: -

Resource risks involve technologies the team has available for their use. Due to costs and other external Constraints, the team may not be able to obtain the needed or best resources to complete parts of the system.

For example, one identified resource is that team members will need laptops with acceptable camera resolution for better and quick testing, but we have.

In addition, the software the team decides to use to keep costs at a minimum, the team is considering open-source software, which is available without charge like opencv library. This is what we need to implement the system correctly, but with user, the system need windows in pc or android in mobile to run with good camera.

This is our resource risk so that you must know what the resources available for you to avoid this risk.

1.3- Functionality Risks: -

Functionality risks have to do with how the system works. Issues that fall under this category include developing a user interface that is not user-friendly or not well-liked by the client, or producing the system function with low percentage of correctness. But we can modify specific parts or functions of the system.

This is the overall function that you must delver in excellent way so that you must work more and more on the percentage of the correctness.

Chapter 4

Snapshots of System

This chapter shows the Snapshots of our system that overview of our system consists of 4 main phases: First starting with signal acquisition to make training data sets, and then extracting useful features of these data sets.
After that classification phase comes by using different classifiers, and finally testing work in different conditions.

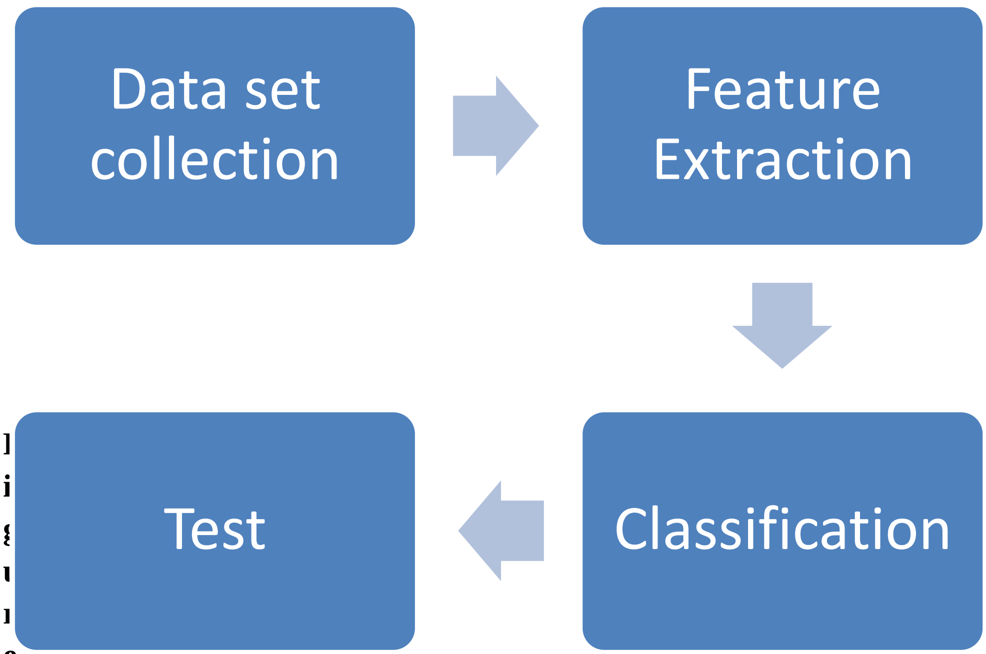
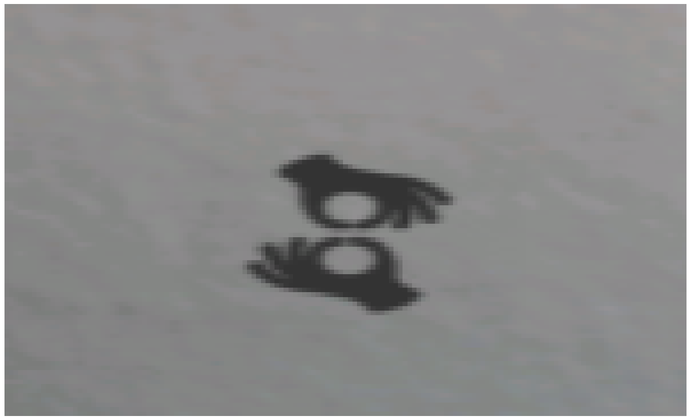


Figure 25 system overview

1. **Mobile application (android)**

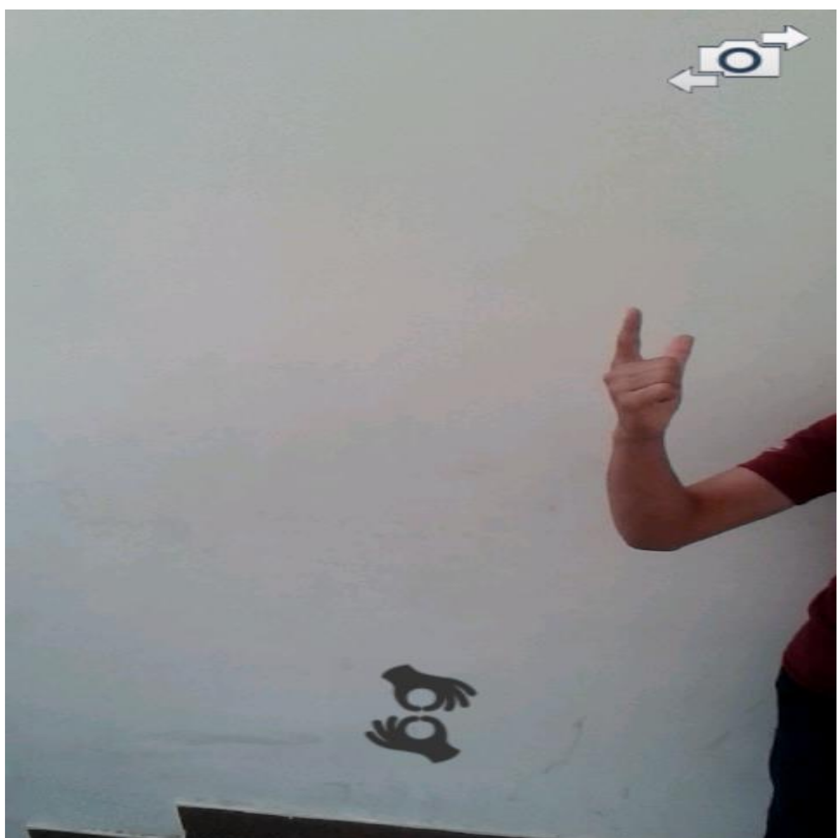
- **At first user opens application**
- **Application will open camera automatically**

- **This button for start program**



- **This button to rotate from back to front or from front to back**

- **Those are number of pictures that express for a word that application will take to make processing in it**
- **The meaning of this word (الله اكبر)**

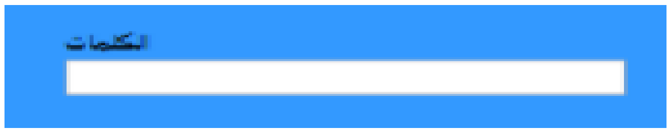


2. **Desktop application with web service**

- **User open application and this is the GUI that will appear**

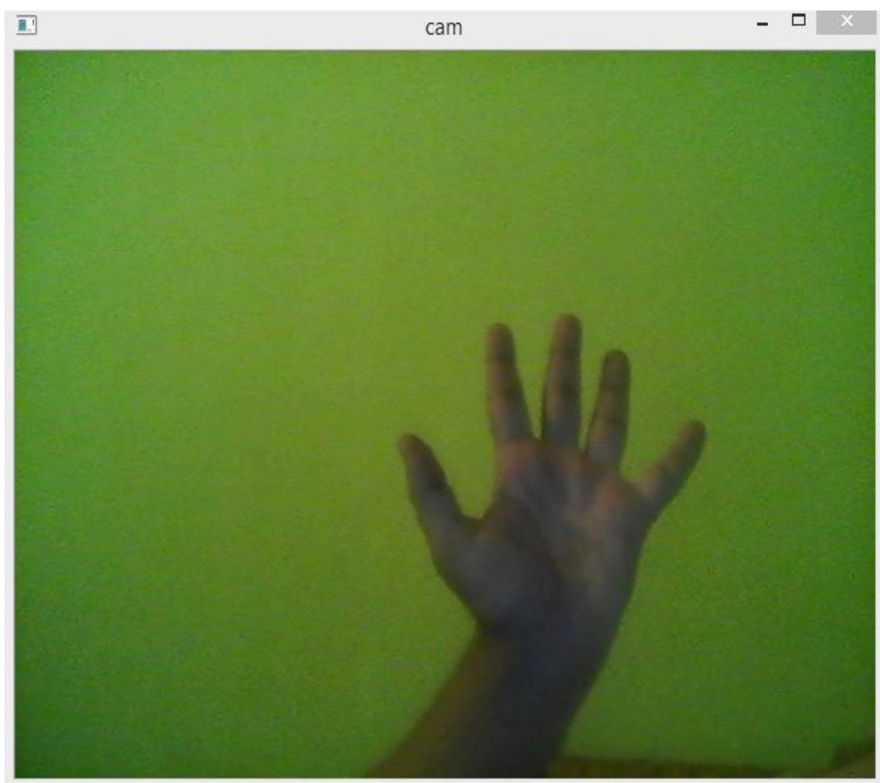
- This button to start opening pc camera and taking pictures

- This label will carry the meaning of word



- This button will stop program

- This express that camera IS opening and taking pictures



Chapter 5

The Result

The accuracy rate of the software was found to be 60%. This figure is lower due to the fact that training was done on the samples of people who did not know sign language and were given a handout to perform the signs by reading from it. So, there was great deal of variation in the samples. Some samples even gave completely wrong readings of the sensors. Testing was also done on the same kind of people.

- **The trained gestures**


صحة


مستقبل

Input- signature	expected	Actual output
18920 7980 2700 3280 1480 2300 2860 0 3060 5040 78360 21680 44160 6180 7000 42120 5020 13900 0 7980 10720 13940 73640 49820 36540 16660 27520 4440 14880 5080 15880 11760 93200 40640 10640 52220 18240 19160 12940 12300 18580 16420 11160 21920 16960 61440 67940 62140 32460 19460 19140 5880 64720 42420 30340 76660 31960 14180 21960 67000 8620 28200 20980 37340 94860 39700 46740 28820 13700 9880 33160 95840 55720 40400 24400 37160 19280 10900 96400 53820 9220 26500 22620 42780	مستقبل	مستقبل

97720 38120 43020 25580 13940 10360 32860 97760 56220 41580 22980 34140 20440 11440 99080 51560		
19220 7000 3380 3380 1500 2900 2960 0 3300 6120 84380 20940 41040 5500 7260 42400 5200 14020 0 7000 9760 17960 68240 53060 35080 15600 26200 4420 14100 4520 20000 17700 89020 43520 12080 51780 17980 19520 10780 10720 17080 14100 10780 23900 14300 63600 68660 62460 32460 18160 17940 6300 69440 43700 31600 73060 29940 14300 21060 69100 8960 24540 22580 39800 98040 42220 46280 22040 13240 10180 34540 95640 60300 39940 24540 31820 20580 11200 98640 54900 8320 25080 22780 40460 96040 40960 47940 22140 14220 9540 32880 95200 58320 40100 24460 31920 21360 12620 96580 51580	مستقبل	مستقبل
17660 5920 2680 2640 1120 2020 1700 0 2400 5700 87060 20680 45720 8940 9200 43120 5300 12360 0 5920 9400 16600 77880 58100 37120 17720 26520 3580 10000 4080 14760 15460 94980 45460 11460 48820 19140 19300 10840 10360 17840 14340 13140 23200 14940 60440 66520 63880 34600 19440 17560 5760 68440 44200 30640 75840 29140 13860 19700 66940 8180 23080 18740 25680 105320 38120 54960 28480 13660 9460 22840 104320 52340 38260 28660 37460 13980 8760 106740 48680 8620 21980 18340 31020 101380 40480 54120 27600 13240 9240 21900 97060 53300 43120 28480 38940 14660 9140 100560	مستقبل	صحه

50300		
18540 7920 3320 5020 2520 3740 5180 0 5700 9400 57800 25100 43960 9080 9780 57220 5740 12800 0 7920 6680 5260 63140 58280 48120 20920 29120 5340 19580 5740 24620 5540 78320 45440 13340 57300 21500 19720 12340 10500 18600 24460 19620 25900 11400 36360 71760 75840 33080 18640 19700 8900 51660 51420 36680 86100 32740 12160 16380 48700 8280 18300 16340 47220 96840 54740 44120 18780 12460 9440 39380 90000 64760 35140 20920 30160 30680 13900 97880 61880 13100 20680 16360 43960 93000 54680 43640 16500 14060 13540 43520 87100 65240 32180 19920 28400 32340 16600 95100 63640	صحه	صحه
18340 5920 2780 4500 2500 3920 4700 0 6560 10760 59420 24520 43680 8760 10620 53960 5720 12620 0 5920 7500 6080 64160 57440 46880 20520 25500 4260 17300 6200 24820 6580 79940 45280 13700 57320 21260 18120 10460 10180 19240 21420 18740 26140 11100 38100 72960 73480 34360 19780 17660 8200 52840 50440 36560 82660 32340 12580 18580 49280 10660 21200 17600 49080 95020 53420 39020 19420 13300 12080 41680 89660 65260 35420 19860 31000 31040 14220 94980 62580 13520 29380 22520 44380 87600 50660 40800 16840 12320 15040 46520 90560 69360 33760 19480 23620 29440 13800 94600 66540	صحه	بسم الله
20520 7460 3520 5020 2860 4740	صحه	مستقبل

5180 0 6300 9820 56500 21760 45220 9580 8680 57240 6020 14500 0 7460 8500 6420 62460 59380 47580 21580 27740 3580 17400 7480 20680 5720 78180 46080 13840 56860 22220 20200 10580 9960 17120 17860 15760 25080 11840 35360 76280 74840 38720 18000 18440 6520 45080 51360 36460 85320 34580 13020 17480 42560 8540 21340 18140 39920 101220 53480 45140 18420 12260 9620 31580 94780 65200 38600 23340 30160 21360 11900 100500 62680 9540 21020 18080 47380 94660 53500 42000 16720 14180 10120 41040 87080 65020 34920 21680 30100 29980 14880 93080 62180		
16640 7720 3260 5740 2640 3260 3740 0 5160 19080 85640 19960 52520 7800 5680 42420 3920 12720 0 7720 7160 53900 56520 72240 34420 15280 20940 4180 14400 3580 22240 41800 67100 53940 18660 44460 16580 14400 10520 19680 15380 18000 10260 15040 31400 65880 66220 73460 24740 15380 15560 27640 71840 32920 36180 80600 24100 10320 39680 75960 10980 21280 15460 18100 94900 36560 64960 42260 15580 11200 16780 95380 49480 35280 38780 48920 14240 7440 99680 48160 9840 21940 18540 35220 98620 43360 48800 27000 14320 10900 33960 99500 59920 37000 24420 32580 21340 11920 102620 56880	بسم الله	بسم الله
16980 7260 3940 5040 2520 3560 3960 0 6400 20620 88040 20180 53540 8260 5740 41980 4240 12740 0 7260 5820 57140 56520 70680	بسم الله	بسم الله

36200 14580 20940 4000 13600 3940 25640 43360 67520 54520 17980 45920 16120 13360 10660 20980 15080 17400 11800 19040 36040 64300 61500 72540 23000 15220 15160 32140 73500 35480 37220 80600 22200 8820 41280 74680 11780 19040 13780 20080 87720 36080 65980 46260 17700 11820 15940 85600 46180 33720 41360 53900 17140 8380 92720 45680 10380 21260 18400 22120 100600 41040 58060 30120 14440 11340 21820 101020 56080 38840 29220 36340 14820 8900 105980 52140		
16440 7840 3440 5120 2260 3620 3860 0 6040 20780 85800 19360 52400 8100 6140 42660 3820 12620 0 7840 8120 57520 54500 71200 34240 15140 21320 4620 12760 3580 26860 42920 64000 52900 19540 45000 17380 14800 10320 22680 16640 20080 16640 22580 35000 63940 49420 67480 24420 17120 18740 31540 77180 35100 28260 75840 22900 9200 41380 75100 10260 22100 18700 32040 98860 44100 49560 25440 15800 11880 31960 97840 58820 36860 23960 32820 19940 11780 103920 56140 10400 20540 18860 30580 99340 43080 51340 26360 15940 11680 30860 97260 58100 37140 24180 34240 20280 11860 103480 54260	بسم الله	صحه
16940 8360 3200 5980 2320 4840 5720 0 8300 24520 86220 21760 52260 7760 5880 41740 4400 12540 0 8360 5840 60620 55100 72180 32100 14960 21460 4580 14540 3880 32500 41980 64540 52120 20080 37980 15780 13620 10600	بسم الله	بسم الله

27340 15580 16580 11980 19840 39500 63340 61840 69140 23140 15580 13900 35820 71820 35760 36560 77240 22820 9500 45820 72040 8840 22160 21360 31520 99480 41260 50900 27620 13700 10060 25520 99800 60760 42100 25120 34840 14820 9020 102520 53640 9040 22940 20440 31080 99700 42400 49280 28080 13940 10220 26840 100600 60120 40400 24740 35140 16100 9340 102240 55000		
6620 52060 40020 53000 22440 10160 18700 5420 20740 8120 42280 23900 44880 39420 14780 28800 11640 10200 10860 37900 15200 26640 16820 10900 48260 37540 54480 45320 24360 16880 24340 55280 55460 26540 29380 52700 15820 8420 56140 52000 17600 15640 37400 54720 69580 25400 27460 22240 52120 26500 46380 69900 33620 17500 43660 38620 44640 13820 75800 36780 11840 25360 69840 49120 59300 16760 23760 34420 76060 16140 42540 59840 27340 23700 71260 41660 39320 11100 60600 36420 11520 25380 69360 51240 59780 16960 25020 34740 75440 17820 42480 59680 26480 23780 71320 40160 38400 10900 60860 36220	اسف	صحه
7620 49280 40480 51600 22760 10560 18560 5320 20380 7980 39780 24360 45640 39180 13680 29980 12800 10640 10660 34440 13640 27460 8520 29220 78420 35920 45020 27860 20440 15480 47020 79620 42600 22060 25700 34660 24720 13560 83880 56280 14380 15680 56400 53200 65600 20440 25780 23540 67140 21800	اسف	صحه

44080 66180 31700 16280 59340 38660 42620 13200 70000 31780 11320 31440 68580 50200 55700 16300 23700 41460 74100 16800 41780 57200 25980 30100 69020 42740 38120 10820 58140 41160 11080 32480 67440 50780 55280 16100 24200 43180 72620 17660 41560 56180 26440 30640 67960 42840 37920 10840 57260 42500		
6540 62420 38300 49660 17840 8580 17440 4040 23140 7500 52360 23620 47020 38620 15740 25280 10700 9480 10400 48420 14120 26380 16240 11560 47980 38920 58200 44420 23620 15860 24620 56020 53860 25460 31560 50560 18420 10360 57000 49920 16740 14440 34440 55340 71420 26800 28780 23180 48980 26460 46960 72180 34580 17240 40440 38940 46240 13320 78320 36360 12780 18920 68760 48440 61860 16860 24900 29440 77160 17420 43460 63860 27920 18500 71860 40600 40800 11680 65160 29380 13420 20780 68740 48980 62380 16540 25160 29380 75780 17820 43400 63600 27420 19180 71700 40720 40680 12240 64820 31440	اسف	اسف
6320 70640 44940 44740 21500 13320 15860 2520 12740 7840 46260 34420 47320 32140 14180 23860 13840 8800 7480 45820 14960 15080 8320 11000 49540 53960 48920 35840 29640 14360 25600 51780 70100 25300 23780 46040 18760 10440 55940 65660 7740 10720 31620 79260 65240 20380 39940 17040 43940 19220 73500 61680 28980 23880 40360 35680 57420 14600 69700 26540 8620 12900 60360 68440 50460	اسف	بسم الله

23360 27640 16160 74220 22620 65700 48220 33900 18680 62700 37320 48780 11340 54020 33020 7580 23940 87540 48120 43940 22620 22900 28620 102160 17780 45340 41680 31740 28780 87960 31300 34320 10560 48200 42560		

- **Conclusion**

The overall tip is to select the fittest algorithm for image processing like pcnn to get signature of the images although this algorithm is not applicable in real time application it take about 16 second to recognize gesture in mobile and 3-5 second in desktop, the classification algorithm must be fit with your application

- This book show all steps analyses, Design and results of our application
- This application help mute deaf person to deal with people
- This application available as mobile application and as desktop application
- This application take number of pictures for deaf mute person that express for a word using Open CV algorithm
- Application make some processing in pictures scaling and graying
- Application take this pictures as input and get signature that express for a word using PCNN algorithm
- Then using machine learning (Feed forward and back propagation) application will now the meaning of word
- Application will print to user and produce a voice that express the meaning of word Desktop application takes 7 seconds and Mobile application takes 15 second

References

[1] Sheng- Yu Peng,KanoksakWattanachote ,Hwei-Jen Lin,Kuan-Ching Li, 'Real-Time Hand Gesture Recognition System for Daily Information Retrieval from Internet ';Fourth International Conference on Ubi-Media Computing 978-0-7695-4493-9/11 2011 IEEE DOI 10.1109/U-IA.2011.45.

[2] Siddharth S. Rautaray,AnupamAgrawal',Real Time Multiple Hand Gesture Recognition System for Human Computer Interaction' I.J. Intelligent Systems and Applications, 2012, 5, 56-64 Published Online May 2012 in MECS

[3] JagdishLalRaheja, RadheyShyam, Umesh Kumar and P Bhanu Prasad, 'Real - Time Robotic Hand Control using Hand Gestures ';in Proceedings of the Inter-national Conference on Machine Learning and Computing, pp. 12- 16, 2010.

[4] Yuvraj V. Parkale 'Gesture based operating system control '2012 Second International Conference on Advanced Computing & Communication Technologies. 978-0-7695-4640-7/12 © 2012 IEEEDOI 10.1109/ACCT.2012.58 .

[5]book of Image Processing Using Pulse-Coupled Neural Networks author is T. Lindblad J.M. Kinser Second, Revised Edition

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms forenergy minimization in vision. submitted IEEE Trans. Pattern Anal. and Machine Intell., 2004.

[7] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In 16th Inter-national Conference on Pattern Recognition., Quebec City, Canada, volume IV, pages 150–155, 2002.