Thesis Report

on

Off-line Recognition of Handwritten Bangla Vowels using Artificial Neural Network

by

Rejwana Rahman (ID: 2010-3-80-021)

Mahmudul Hasan (ID: 2010-2-80-028)

Submitted to

Department of Electrical & Electronic Engineering
Faculty of Science and Engineering
East West University

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical &Electronic Engineering
(B.Sc. in EEE)

Fall, 2015

Approved by

Supervisor	Chairperson
Muhammed Mazharul Islam	Muhammed Mazharul Islam

Declaration

We, hereby certify that our thesis work solely to be our own scholarly work. To the best of our knowledge, it has not been shared from any source without the due acknowledgement and permission. It is being submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronics Engineering. It has not been submitted before for any degrees or examinations of any other university.

-------------------------------- -----------------------------

Mahmudul Hasan Rejwana Rahman

Abstract

Recognizing handwritten characters is a major challenge in the field of pattern recognition. A unique solution is difficult to find because of the high variability in the samples.

Although a lot of research is going on to identify Bangla handwritten characters, the developed methods cannot be universally applied due to the lack of a central database. Each researcher has to manage their own data, which creates non-uniform results.

Artificial Neural Network (ANN) is a well-established method in the field of pattern recognition for recognizing handwritten character. Various ANNs have been developed to identify handwritten characters in different languages. Here, we use ANN with back propagation learning algorithm to classify Bangla vowels.

The data used to identify the characters was collected by us and was fed to the neural network after preprocessing.

Using the pixel values as data, the best result obtained was 68.9% recognition rate for 16 hidden layers, which is quite poor. To improve the results, we used a Gabor filter to extract directional features from the character images.

With the feature data, the best result obtained was for 207 hidden layers, which is 79.4%. Finally, the drawbacks and future works are briefly discussed.

Acknowledgment

First and foremost, we would like to express our sincere gratitude to our supervisor, Dr. Muhammed Mazharul Islam, Assistant Professor, Department of Electrical and Electronics Engineering, East West University for guiding us during our research. We are also very grateful to our parents, advisors, other faculty of the department, and our friends for their support and encouragement. We also thank the almighty Allah for giving us the patience to complete this task.

Authorization

We hereby certify that we are the sole authors of this thesis. We authorize East West University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Mahmudul Hasan

Rejwana Rahman

We authorize East West University to produce this thesis by photocopy or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Mahmudul Hasan

Rejwana Rahman

Table of Contents

List of Figures

List of Tables

Ch 1: Introduction

Handwriting recognition is the automatic identification of handwritten text using a computer and interpret into a machine-based textual representation [1].Handwritten character recognition is a challenging task and one of the most active research areas in the field of pattern recognition and image processing[2].

The recognition process should be robust and source independent. The sources of handwritten characters can be many, e.g. paper documents, photographs, touch-screens and other devices. The input is generally in the form of images which are processed to obtain the machine interpretation. The recognition process can be “off line” or “on line”. Off line recognition involves the recognition of characters obtained from images, while on line recognition involves sensing the tip movements of a suitable device where direct handwritten text input is possible (e.g. tablets, smart phones etc.).

The process of character recognition is ideally a component of optical character recognition (OCR). However, the complete system also contains other functions such as segmentation (i.e. separation of words from a sentence), preprocessing etc. The recognition is performed through a classifier (i.e. algorithm). The performance of character recognition systems largely depends on the feature extraction approach and the classification/learning scheme. Although many approaches have been studied for feature extraction, directional features are one of the most efficient for handwriting recognition[2].

1.1 Necessity of Handwritten Character Recognition

The main applications of character recognition can be found in reading bank checks, postal addresses, paper forms, many types of older paper documents for record keeping to create a searchable database etc. Considering its varied necessities, researchers have developed techniques to recognize different handwritten scripts such as Arabic numerals, Latin alphabets, Japanese Katakana and Kanji (Japanese version of Chinese) characters, Chinese characters, and Hangul characters. Work is still being done to improve such techniques and recognize Greek, Indian and Arabic scripts [3].

1.2 Available Methods

Many different recognition techniques are currently available for handwriting recognition[4]. These techniques can be divided into four categories, i.e. template matching, statistical techniques, syntactic techniques, and neural networks [1], Template matching determines the amount of similarity between two groups of shapes. Matching techniques include direct matching [5] and deformable templates and elastic matching [6].

Statistical techniques use statistical decision functions and determine the probability of the observed pattern belonging to a certain class based on a set of optimal criteria. There are several popular approaches belonging to this category, some of which are mentioned below:

- The k-Nearest-Neighbor (k-NN) method is a popular recognition method, where a character is classified based on the patterns in its neighborhood in a feature space. This method suffers from a high computation cost, and researchers are working to develop faster computation techniques for handwriting recognition [7].
- In the Bayesian classifier method, a posterior probability assigned to a class of patterns and the conditional probability density function for a feature vector is estimated [8].
- Support Vector Machine (SVM) are binary classifiers that uses quadratic optimization technique in a high dimensional feature space to classify patterns [9]. For multi class classification, SVMs can be combined. A high computation cost is the main disadvantage of the method.

In syntactic methods a hierarchical approach is used where patterns are thought of as composed of simplest sub-patterns called primitives. Character patterns are then represented by the inter-relationships among primitives. The patterns are then related to the syntax of a language, and grammatical rules are used to identify the handwritten script [1].

An Artificial Neural Network (NN) is a computing system with many parallel interconnections of simple processes [10]. ANNs have many advantages over other methods, i.e. self – adaptability, which ensures good performance with noisy data, the ability to approximate any non-linear function, the possible of parallel implementation and the ability to handle large databases efficiently[11].

ANNs are widely used in pattern recognition and have produced promising results in handwriting digit recognition. Multi-Layer Perceptron (MLP) is the most widely used neural network[12]. An MLP architecture with back-propagation training algorithm is the most frequently used classifiers for handwriting recognition [13].

For Bangla basic and compound characters, a comparative study was performed between ANN with MLP and SVM, with an accuracy rate of 79.25% with ANN[14]. Another research used

MLP ANN and obtained 95% recognition for only the letter “ক”[15].On the whole character set of Bangla alphabet, one team reached 84% accuracy using ANN [16].

1.3 Difficulties in Handwritten Character Recognition

One of the main problems is the deformation of the characters [17]. Handwriting recognition is difficult to perform due to the variability among writings of different persons. The visual appearances of the same characters differ greatly due to the writing characteristics of different persons. Therefore, a recognition system has to learn how to distinguish these character instances from another. A way to solve the problem is to find and use features that are invariant to the deformations.

1.4 Objective of the Thesis

The objective of the thesis is to develop an artificial neural network to classify Bangla vowels. The method uses two different types of feature extraction techniques and compares the positive recognition rates of the systems.

1.5 Structure of the Thesis

The thesis is structured as follows: Chapters 2 covers the fundamentals regarding handwriting recognition, and Chapter 3 discusses the results. Finally, the conclusion follows in Chapter 4.

Ch 2: Proposed Method for Character Recognition

2.1 Outline

A problem to reproduce benchmark recognition systems is the lack of a large universal database on Bangla characters. Therefore, to develop a system, the first step is to gather sufficient data samples. We collected data samples ourselves and created a small database of Bangla characters. As a standard database on Bangla characters were not available, this step was necessary.

The data was collected in pre-designed paper forms. After the forms were filled up by volunteers, each page was scanned and the image was stored in the computer for processing.

Next, individual handwritten characters are picked up from the scanned images and saved under separate names. Each separate character is then resized to its bounding box. The bounding box of each character is different, so, the size of each image are different.

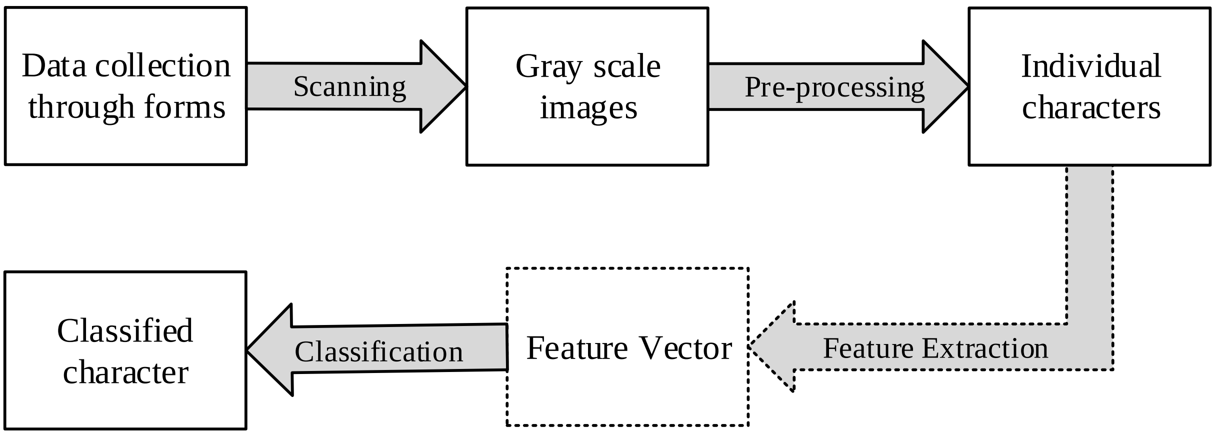


Figure 2.1: Flowchart for character recognition

Since the character images were in gray scale, they were converted to binary. The characters were then size normalized and an artificial neural network was used for recognition the characters. Figure 2.1 shows the outline of the method.

Although we collected data on both vowels and consonants, here we only worked on the vowels. The consonants will be useful in our future works.

2.2 Data Collection and Pre-processing

The data was collected in pre-designed printed forms from a total of 95volunteer students of East West University. Figure2.2 shows a sample of the data collection forms. The volunteers were given the forms and were asked to write the different vowels and consonants in the specified boxes. They were also instructed to write the characters in such a way that the strokes do not cross out of the boxes.

The boxes were primarily used so that the boundaries could be used to easily separate the handwritten characters later on.

After the forms were collected, each of these is scanned in gray scale at a resolution of 300 dpi (dot per inch). This resolution was selected because it reduced the scanning time and produced fairly noise free images. Since image clarity is always helpful to obtain a good recognition performance, the next higher resolution setting of 600 dpi should have been more suitable, but since documents are usually scanned at 150 to 300 dpi, we did not use the higher resolution setting.

Furthermore, we found that for our case, scanning with 150 dpi resolution produced too much noise and the images were of poor quality, which would affect the recognition performance. So, we chose to scan at 300 dpi resolution.

After scanning, the images were stored in the personal computer (pc) as PNG (portable network graphics) files. The PNG format was used because they use a lossless data compression technique, which reduced the memory space to store the file on computer [18].

Next, we used a program to identify the boxes by searching for the horizontal and the vertical lines of the scanned forms. This is done by applying a Sobel filter to detect the edges in the image [19].

The filtered image contains a binary value of 1 along the edges and a value of 0 otherwise. The box boundaries were easily selected from the filtered image and the handwritten characters were separated from the scanned form. Each handwritten character was stored in the pc under a separate name. Figure 2.3 shows samples of the extracted characters from the data collection form.

Since the images were scanned in gray scale, the characters were converted to binary using Otsu's method [20]. This is a standard method used by researchers to convert images to binary.

The dark part (i.e. background) having a value 1 and the white part (representing the character) as 0 (shown in the figure). All the characters were also size normalized for ease of recognition to 27x26 pixels, as shown in Figure 2.4.

অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	
ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	
ট	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন	
ট	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন	
প	ফ	ব	ভ	ম	য	র	ল	শ	ষ	
প	ফ	ব	ভ	ম	য	র	ল	শ	ষ	
স	হ	ড়	ঢ়	য়	ৎ	◌ং	◌ঃ	◌ঁ		
স	হ	ড়	ঢ়	য়	ৎ	ং	ঃ	ঁ		

Figure 2.2: Data Collection Form

In this way, we obtained 95 sets of vowels for using in the artificial neural network.

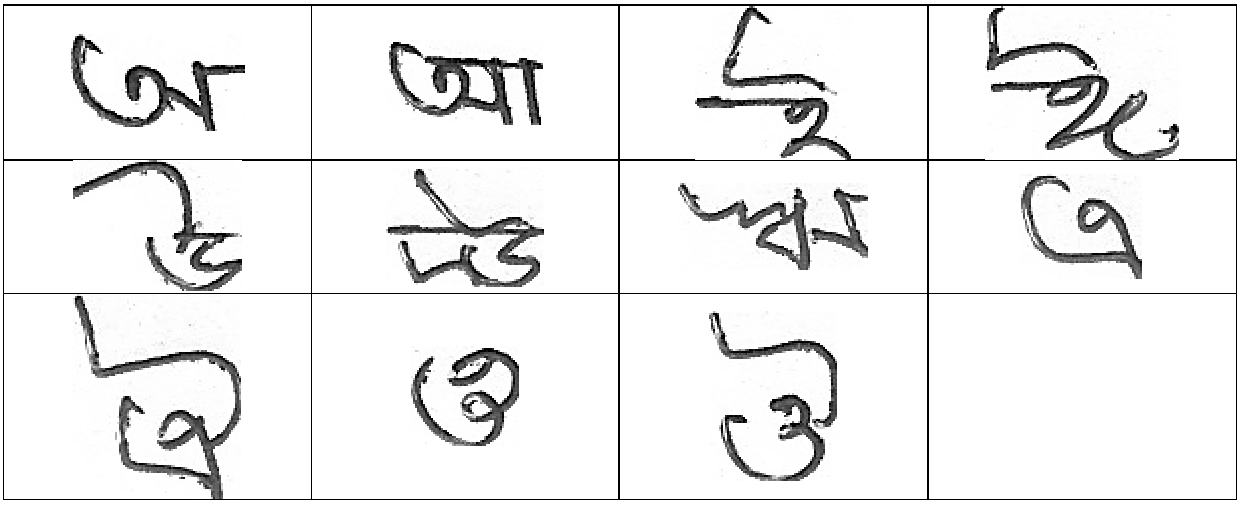


Figure 2.3: Extracted character images from the data collection form

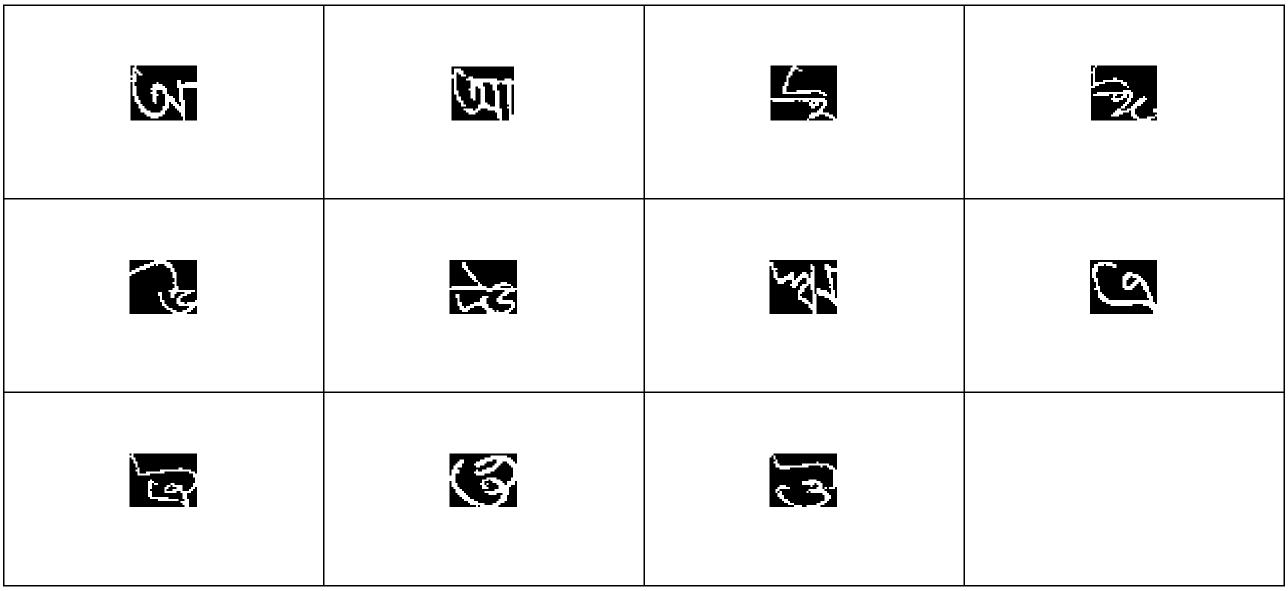


Figure 2.4: Size normalized binary images (27x26 pixels)

2.3 Artificial Neural Network (ANN)

2.3.1Structure of ANN

We use ANN as classifier to identify the characters. The network structure is modeled on the interconnections of neurons in a human brain, and consists of nodes in different layers with inter connections [10]. Figure 2.5 shows the structure of an artificial neural network.

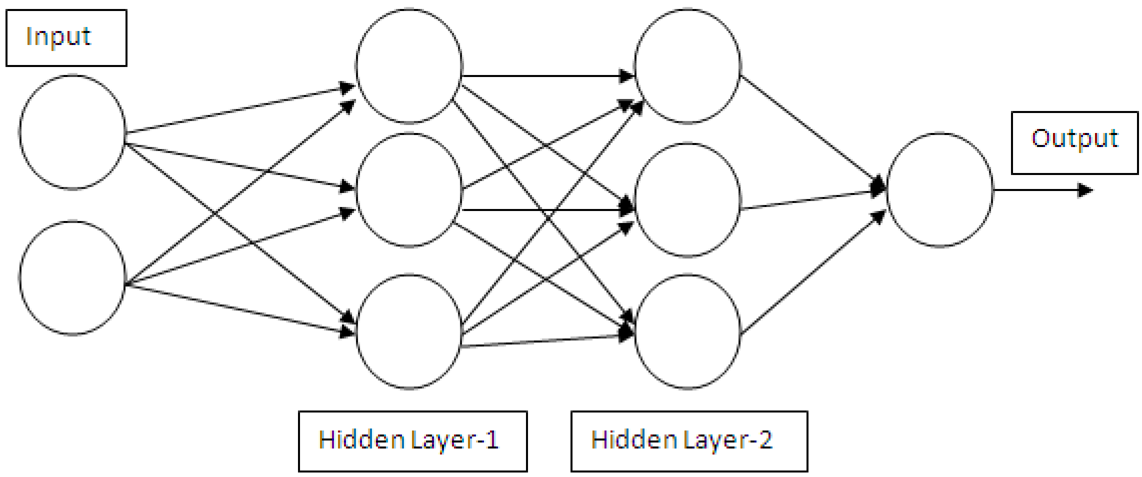


Figure 2.5: Structure of an Artificial Neural Network

- **Input Layer:** Input is the given data on which a neural network will work on. In our case, the character samples are the input. We have 95 samples each of 11 characters (i.e. a total of 1045characters) as input. We use 27×26 pixel for each letter which gives 702 input nodes to the network for each character.
- **Output Layer:** The output is the class to which a character belongs. The preciseness of the output of an ANN depends mostly on how the network has learned. A network has to be trained with some samples in order for it to identify unknown samples. Learning basically is an algorithm in order to determine a mapping from the dataset to the corresponding class labels. Three major types of learning are there *supervised learning*, *unsupervised learning* and *hybrid learning.* Depending on the training the output tries to identify the most accurate desired output.
- **Hidden Layer:** Hidden layers are a number of middle layers considered as nodes between input and output of an ANN. Although the number of hidden layers depends on the system, input and desired output of the system, an ANN cannot learn without any hidden layers. Generally two layers are used for a simple recognition system. If there is no hidden layer used in the network, it will be capable of only representing linear separable functions or decisions. One hidden layer can approximate any function that contains a continuous mapping from one finite space to another. Two hidden layers represent the arbitrary decision boundary to arbitrary accuracy, which leads to approximate smooth mapping to any accuracy [12].

  The hidden layer is a very important part of deciding the overall neural network architecture. The major portion of the network consists of hidden layers (and neurons).

Too many neurons in the hidden layers does not necessarily has a positive effect on the recognition rate. Several problems may occur due to the excessive neuron number compared to its input. One important problem is over-fitting, which generally occurs when a network model is excessively complex due to many parameters. A model which has been over-fit will generally have poor predictive performance. Another problem occurs due to too few number of neuron which is called under fitting. So the maintenance if neuron in the hidden layers is very important.

To decide on the optimal number of hidden layers, we need to check the results for different layers starting from single layers.

2.3.2 Computational Model of Neurons

Mathematically, the output of a neural network can be defined as the weighted sum of *n* input signals, $x_j,\ j = 1, 2, \ldots, n$ such that

$$y = \theta\left(\sum_{j=1}^{n} w_j x_j - u\right) \tag{1.1}$$

Where $\theta(\cdot)$ is the activation function, w_j is the weight associated with the *j*-th input. Therefore, the output is 1 if $\sum_{j=1}^{n} w_j x_j > u$ and zero otherwise.

2.3.3 Network Architecture

Based on how the neurons are connected, ANN can be categorized into two types:

1) Feed-Forward Network
2) Recurrent Network

1) <u>Feed Forward Network:</u>

As the name implies, feed forward networks do not have any feedback paths. In a feed forward network, information passes from input to output directly.

2) <u>Recurrent network:</u>

Recurrent networks are a type of neural network that has loops or feedback connections. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.

Multilayer Perceptron (MLP):

A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps a set of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron acts as a nonlinear activation function and utilizes a supervised learning technique called 'back propagation' for training the network. Ultimately a perceptron having many layers which is called MLP is a developed version of neural network.

2.3.4 Learning Algorithm and Back Propagation:

Back propagation is a popular learning algorithm for pattern recognition, dynamic modeling, and sensitivity analysis. It is also used to fuzzy logic, fluid dynamic model and other logical modeling. This is a powerful method for modeling for supervised learning network. Figure 2.6 shows a three layer feed-forward neural network.

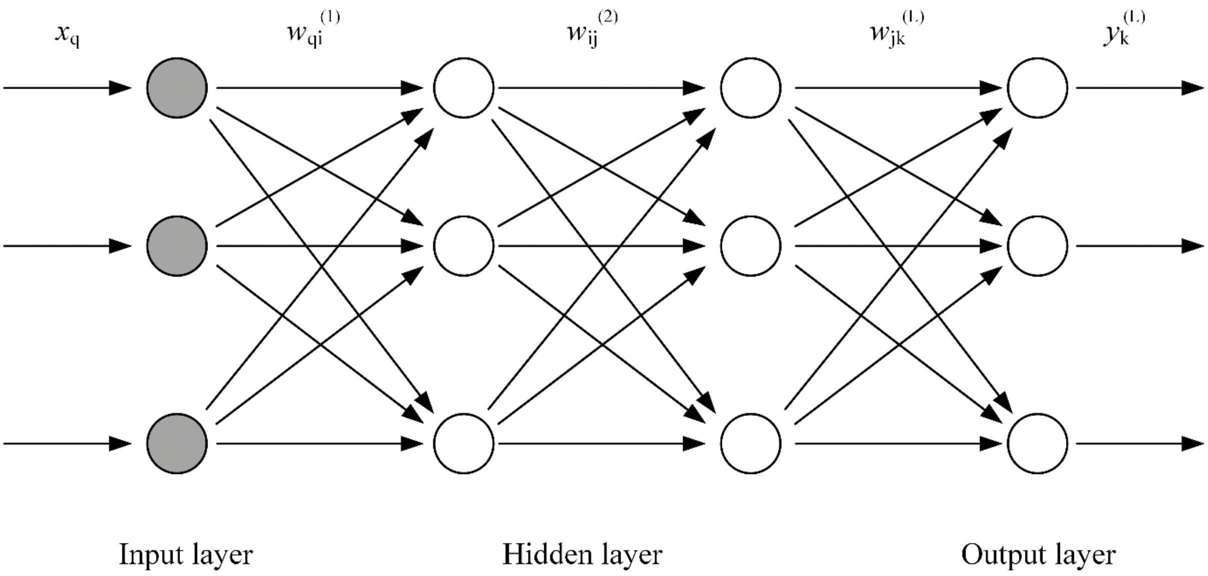


Figure 2.6: A three layer feed forward network

If w_{ij} is the weight on the connection between the *i*-th unit in layer (*L*-1) to *j*-th unit in layer *L*, $\{(\mathbf{x}^{(1)},\mathbf{d}^{(1)}),(\mathbf{x}^{(2)},\mathbf{d}^{(2)}),\ldots(\mathbf{x}^{(p)},\mathbf{d}^{(p)})\}$ are a set of *p*- number of input-output pairs (called training pairs), where each input $\mathbf{x}^{(L)}=[x_1^{(L)},x_2^{(L)}\ldots x_n^{(L)}]^T$ is a vector in *n*-dimensional space, and each output $\mathbf{d}^{(L)}=[d_1^{(L)},d_2^{(L)}\ldots d_m^{(L)}]^T$, where $d_q^{(L)}\in[0,1]$ is a vector in *m*-dimensional space. For classification purposes, *m* is the number of classes. The cost function is then defined as

$$E = \frac{1}{2}\sum_{i=1}^{p}\left\|\mathbf{y}^{(i)} - \mathbf{d}^{(i)}\right\|^2 \qquad (1.2)$$

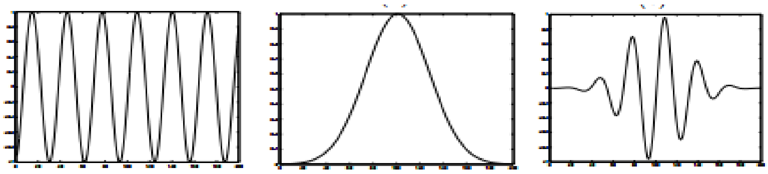
The back-propagation learning algorithm is used for determining weights in a multi-layer perception. The back-propagation algorithm is a gradient-descent method to minimize the squared-error cost function [21].

2.4 Feature Extraction using Gabor Wavelet Transform

Features are most representative information related to data which reduces the variability among patterns of the same class while increases the variability among patterns of different classes. Features are extracted to better distinguish the classes and are application domain specific. Features can be broadly classified into two categories, i.e., statistical features and structural features [22].

Statistical features are pixel density, mathematical transformation etc. Structural features include strokes, contours etc. Statistical features are easily affected by the deformation of symbols, so structural features are more suitable for handwritten character recognition.

Gabor filter is useful for extracting directional features of a character. Gabor wavelets (or filters), named after Dennis Gabor, are linear filters used for edge detection in image processing. A two dimensional (2D) Gabor filter is sinusoidal plane wave modulated by a Gaussian kernel function[23]. Spatial orientation and frequency of Gabor filters are similar to the human visual systems. Thus, image analysis with Gabor filters is similar to as being perceived by the human visual system. Figure 2.7 illustrates a Gabor wavelet in one dimension.



(*a*) (*b*) (*c*)

Figure 2.7: (a) Sinusoid signal (b) Gaussian kernel function (c) Gabor wavelet

The Gabor wavelet in two dimensions can be expressed as

$$g(x,y,\lambda,\theta,\psi,\sigma,\gamma)=\exp\left[-\left(\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\right]\sin\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

Where, $x'=x\cos\theta+y\sin\theta,\ y'=-x\sin\theta+y\cos\theta$

Here, *x*, *y* are spatial coordinates that specify the pixel value of an image at coordinates (*x*,*y*). Each "pixel" in the transform tell us the "intensity" of such a wave. Pixel is defined as a minute area of illumination on a display screen. Basically pixel is a dot on a given grid not the size of the grid.

Again, λ represents the wavelength of the sinusoid; θ represents the orientation of the Gabor function. Ψ is the phase offset of the sinusoid. The phase offset is symmetric when$\psi = 0\ or \pi$. The phase offset is anti-symmetric when $\psi = \frac{\pi}{2}$ or - $\frac{\pi}{2}$.

Here σ is the standard deviation of the Gaussian envelop. This depends upon the spatial frequency to be measured. The standard deviation of Gaussian factor determines the size of the respective field, and γ is the spatial aspect ratio.

Ch 3: Results

We collected data from 95 people, so for the vowels, considering 11 characters for each people, we have a total of 95 x 11 = 1045 characters. For classification, the artificial neural network (ANN) has to be trained with a set of data. The classification performance (or accuracy) is then tested with another set of disjoint data set. The training data is randomly chosen from the data set. While choosing, it was made sure that each character class had equal number of representatives in the training set. This was done to avoid biasing the system towards or against a particular character.

3.1 Without Feature Extraction

We used 80% of the total sample data for training the system, and the rest of 20% for testing the system. We built a neural network using MATLAB and trained the system first to determine the neuron weights. The system with the fixed weights was then used to test the classification performance on the test data. Figure 3.1 shows a graphical representation of a neural network with 207 hidden layers created in MATLAB.

The number of classes in our case is 11, and each class must be uniquely identified with a vector of the size of the class size (i.e. 11). The vector should have all elements equal to ‘0’ except for the class number position, which should have a ‘1’. Table 1 shows the class representations for all the classes for the system.

Each character image of the size of 27 x 26 = 702 pixels is rearranges into a vector and fed directly to the network. We changed the number of hidden layers from 10 to 500 in spans of 30 and tabulated the classification performance in Table 2.

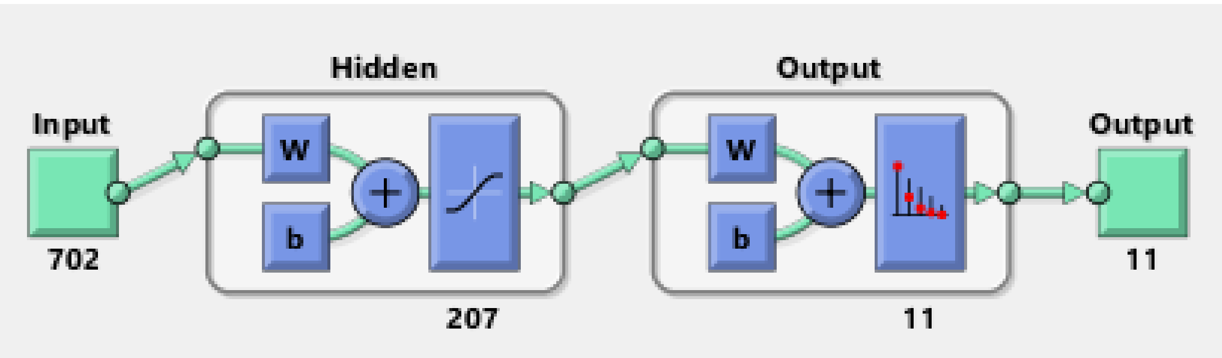


Figure 3.1: A neural network created in MATLAB

Table 1: Class representation for neural network

Class number	Character	Output classvector representation
1	অ	10000000000
2	আ	01000000000
3	ই	00100000000
4	ঈ	00010000000
5	উ	00001000000
6	ঊ	00000100000
7	ঋ	00000010000
8	এ	00000001000
9	ঐ	00000000100
10	ও	00000000010
11	ঔ	00000000001

Table 2: Classification performance for different hidden layers (without feature extraction)

Number of hidden layers	Classification accuracy in %
1	15.8
16	68.9
30	67
60	64.1
90	57.4
120	64.1
150	50.7
180	58.9
210	52.6
240	48.8
270	52.6
300	56.9
330	60.8
360	51.7
390	48.8
420	49.3
450	46.9
480	60.3
500	60.8

Figure 3.2 shows a plot of the classification performance against number of hidden layers. The best results of 68.9%are obtained for 16 hidden layers.

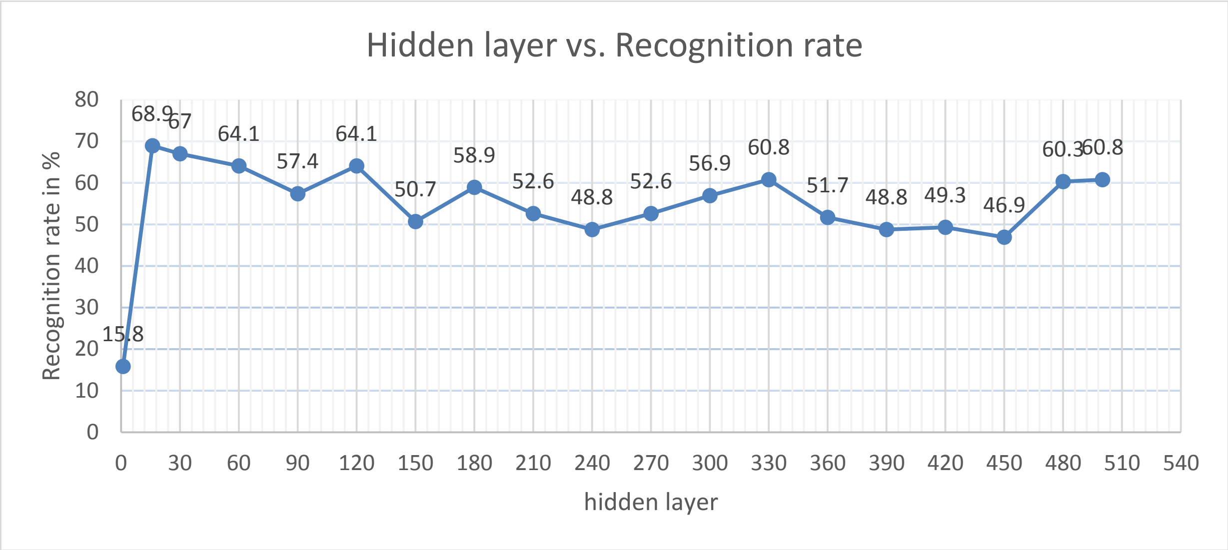


Figure 3.2: Classification performance against number of hidden layers

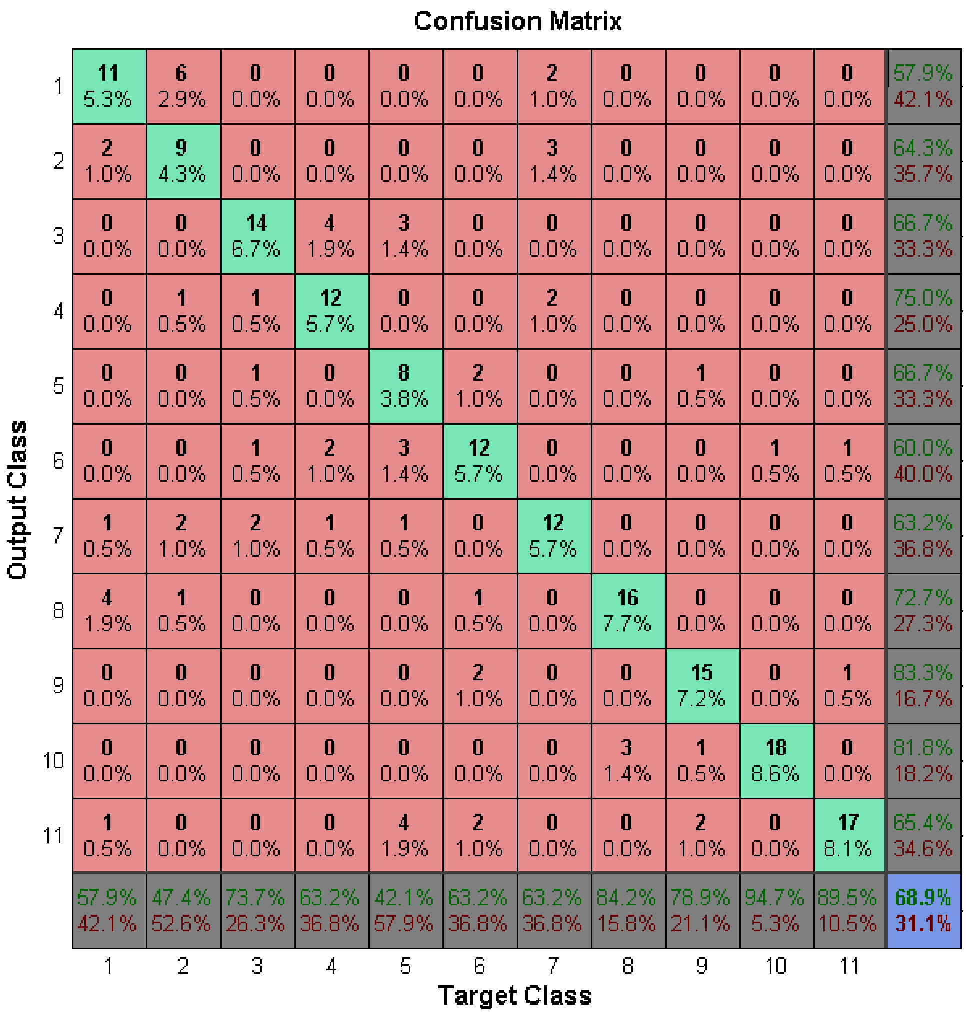


Figure 3.3: Confusion matrix for 16 hidden layers in ANN without feature extraction

Figure 3.3 shows the confusion matrix of the network output for 16 hidden layers. Each class has nineteen samples. The worst misclassification (42.1%) occurs for class 5 character

3.2 With Feature Extraction

To increase the classification performance, we next used a Gabor filter to extract four directional features from each of the characters. The parameters used for Gabor filter are, γ (aspect ratio) = 0.5, ψ (phase shift) = 0, θ (orientation) = [0°, 4°, 90°, 135°]. The four orientations are shown in Figure 3.4.

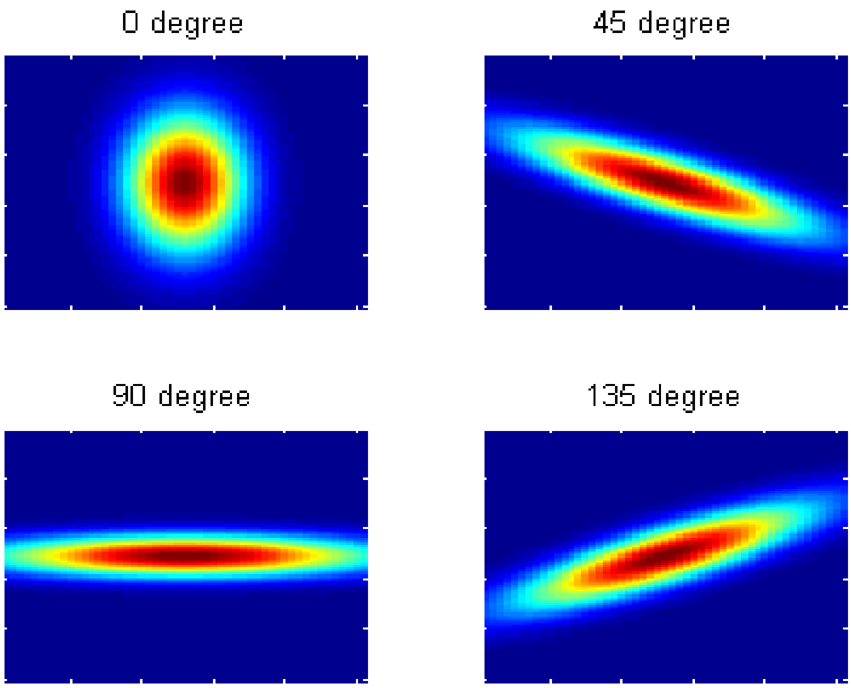


Figure 3.4: Spatial orientations of the Gabor wavelet

If the wavelength of the Gabor wavelet is set too large, the filtering has no affect. The same thing happens if the wavelength is set very small. Therefore, the size of the wavelength must be shorter than the sample image size but not too small. For most of the cases, the value is chosen heuristically. In our case, since the image size is 27x26 so the size of the Gabor wavelet must be kept smaller than the image size and here it is determined as 9x19.

Figure 3.5 shows three dimensional representation of the Gabor filter for different orientations. Figure 3.6 shows a sample character and Figure 3.7 shows the directional features after filtering.

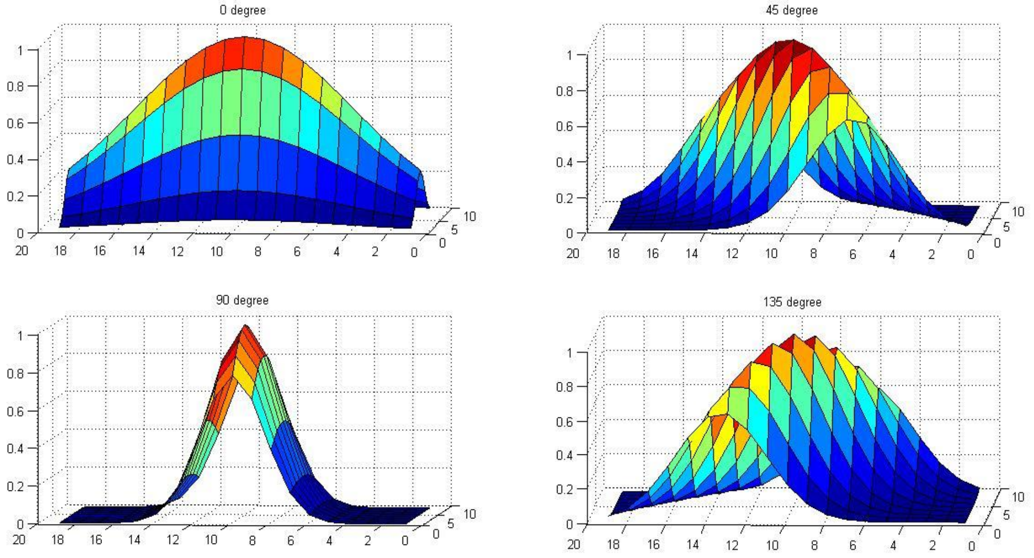


Figure 3.5: Three dimensional representation of the Gabor filter for different orientations

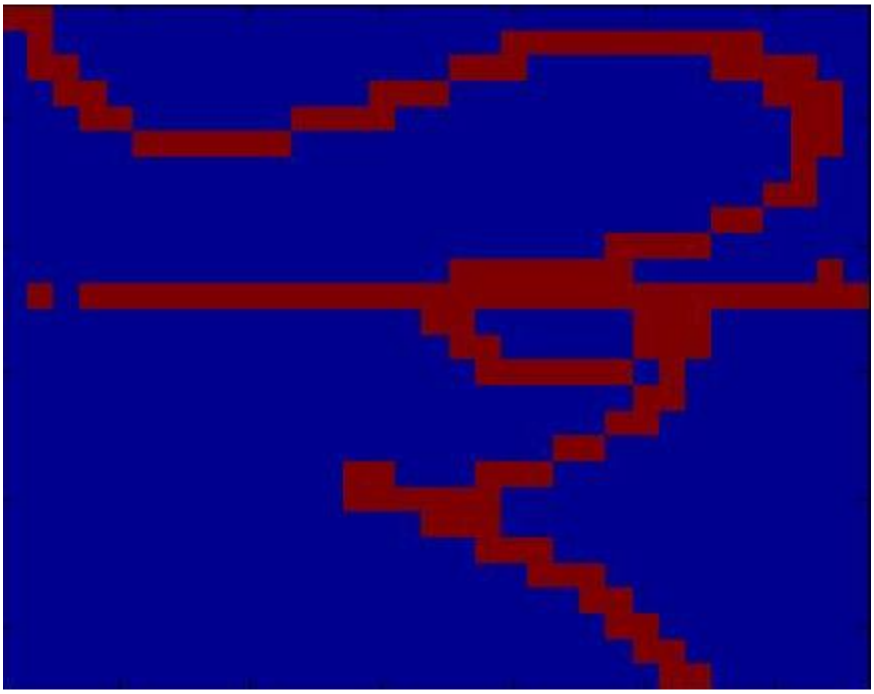


Figure 3.6: Original shape of the character of class no.3

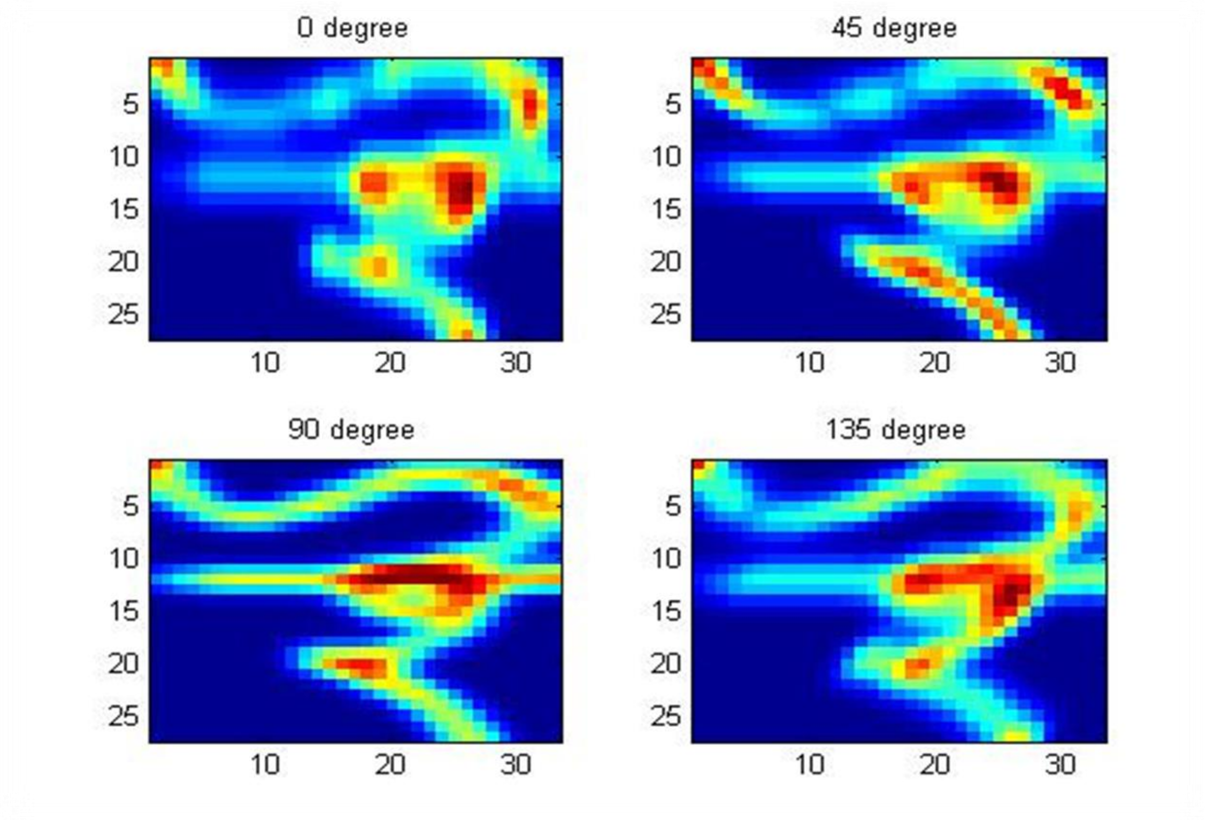


Figure 3.7: Filtered character

Table 3: Classification performance for different hidden layers (with feature extraction)

Number of hidden layers	Classification accuracy in %
1	17.2
30	73.7
60	72.7
90	70.8
120	70.8
150	69.4
180	73.2
207	79.4
210	70.3
240	69.4
270	71.3
300	72.2
330	70.3
360	67.4
390	69.4
420	65.6
450	71.8
480	65.1
500	67.9

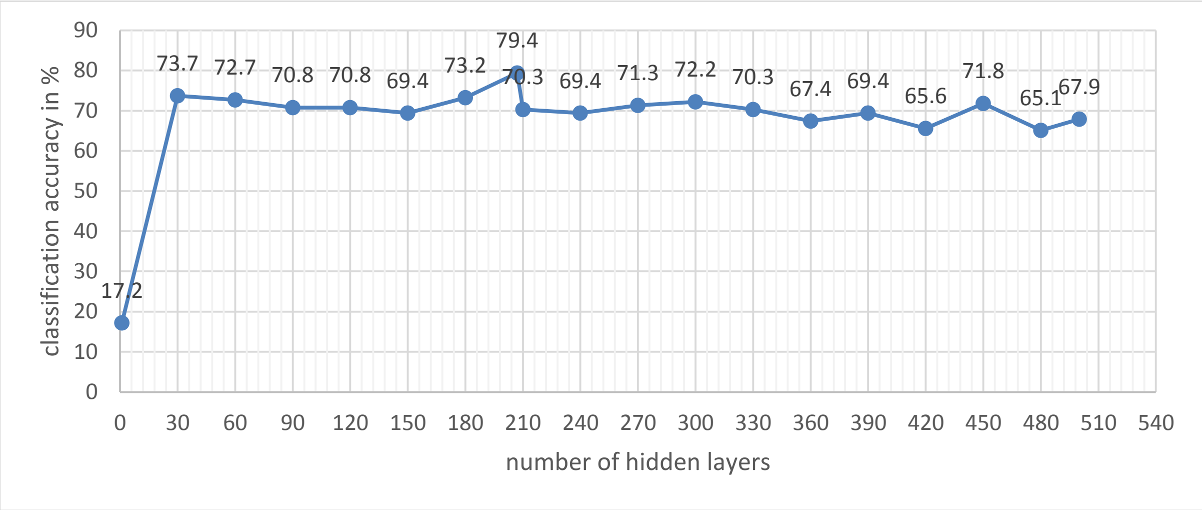


Figure 3.8: Classification performance against number of hidden layers

(With feature extraction)

Table 3 shows the classification accuracy against the number of hidden layers. Figure 3.8 shows the results of table 3in graphical form. With the extracted features, the maximum classification accuracy occurs for 207 hidden layers. Figure 3.9 shows the confusion plot for the best result. Here it seems that the 2 no. index represents ‘আ’ is the best recognized characters in this constructed Network. Here 100% characters (19 out of 19 character sample) are recognized perfectly. 11 no. index represents ‘ঔ’has the accuracy rate of 89.5% (17 out of 19 character sample). 6, 8, 9 and 10 no. index represents ‘ঊ’,‘এ’,‘ঐ’,‘ও’ has84.2% accuracy as it can recognise 16 characters out 19 test sample.1 no. index represents ‘অ’ has the accuracy rate of 73.7% as it recognize 14 characters out of 19 characters. 3,4,5 and 7 no. index represents‘ই’, ‘ঈ’,‘উ’,‘ঋ’has worst recognition rate of 68.4% . Here only 13 characters are recognized perfectly out of 19 sample test characters.

From the confusion matrix, it is seen that two ‘অ’ samples are misclassified as ‘এ’. another three are recognized as ‘আ’,‘ঋ’, and‘ঔ’. Three ‘ই’sample is misclassified as ‘ঊ’, two recognized as‘ঈ’and another one as ‘ঐ’. Three ‘ঈ’ samples are misclassified as ‘ঊ’.Two recognized as‘ই’and another one is as‘উ’.

Two samples of‘উ’are misclassified as ‘ঊ’, and another four are misclassified as‘ই’,‘ঋ’,‘এ’, and ‘ঐ’. From the sample of ‘ঊ’, two is misclassified as‘উ’ and another one seems to be misclassified as, 'ঐ'. From the sample of 'ঋ'two classified as 'আ' another two as 'উ' and the rest two misclassified as 'অ' and 'উ'. From the 8 no. index sample 'এ' misclassified three characters as 'ই', 'ঐ' and 'ও'. Three test samples from the character 'ঐ'is misclassified. Two as'ঔ'and one as 'ঊ'. 10 no. character index'ও'has three misclassification as'ই','উ' and 'ঔ'

The last character 'ঔ' has two misclassifications as 'উ' and one as 'ঐ'.

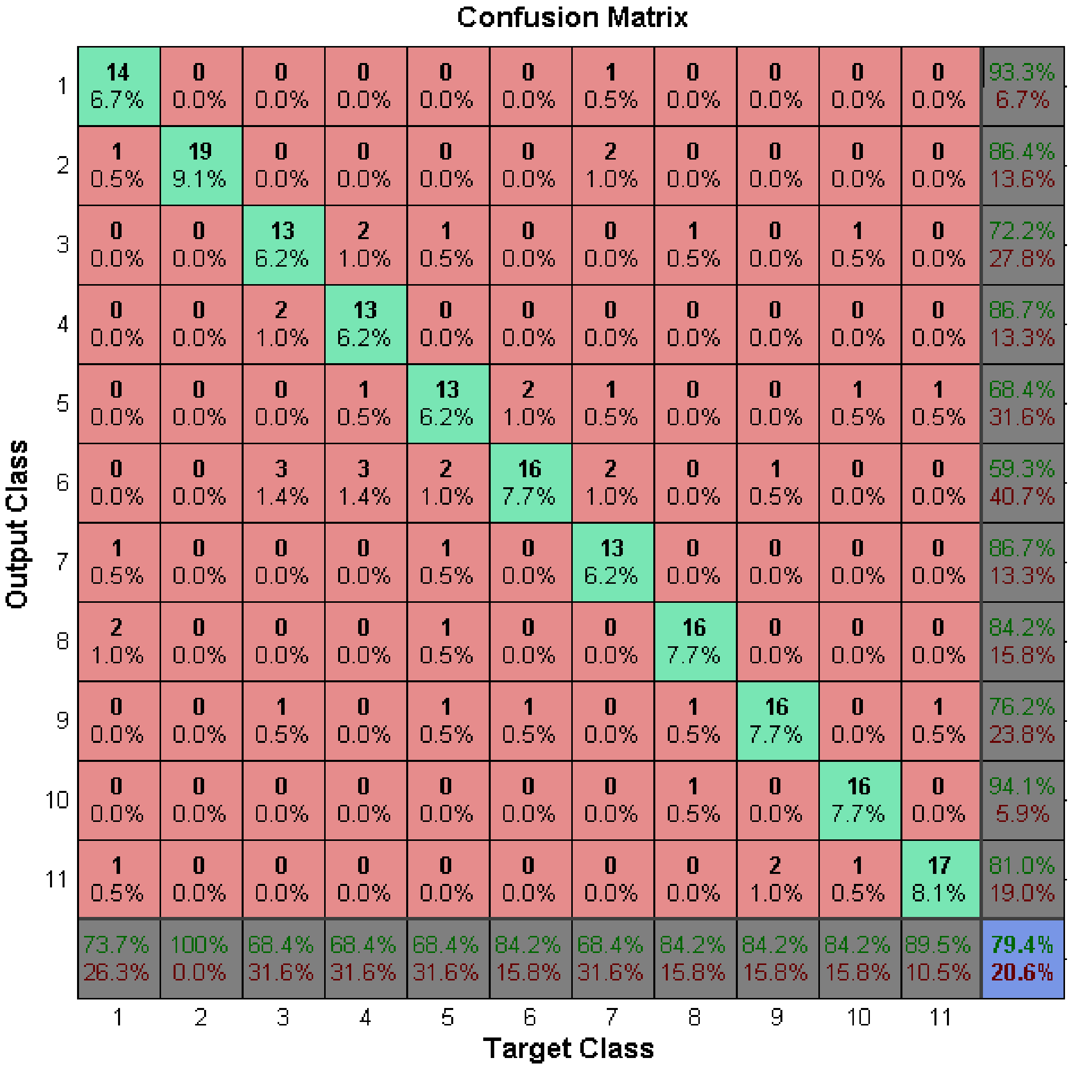


Figure 3.9: Confusion matrix for 207 hidden layers in ANN with feature extraction

3.3 Misclassified Samples

Some of the misclassified samples are shown in Figure 3.10and Figure 3.11for 207 hidden layer. The first number above each figure is the actual (or target) class number, while the second number is the identified class number.

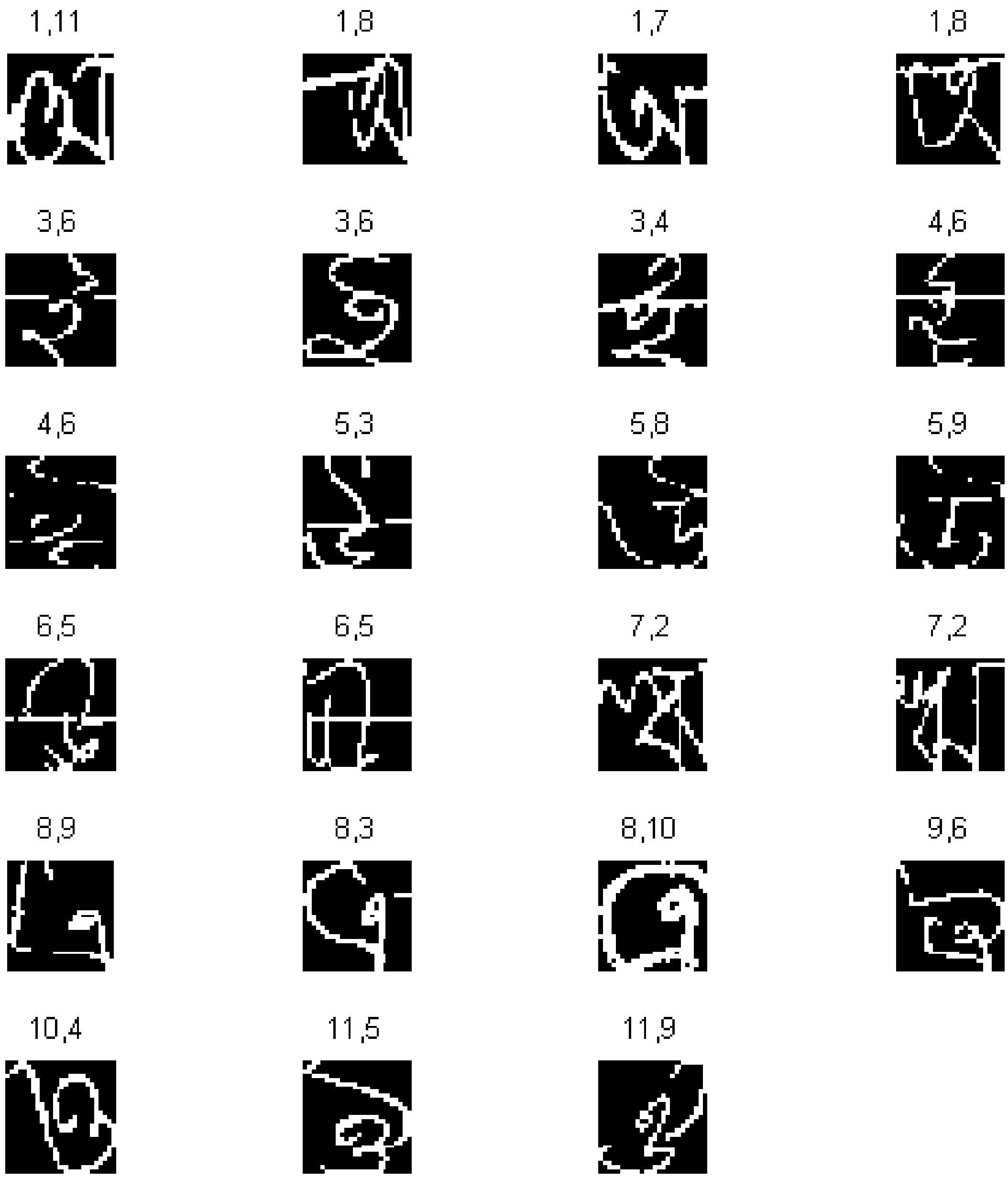


Figure 3.10: Misclassified characters (part 1)

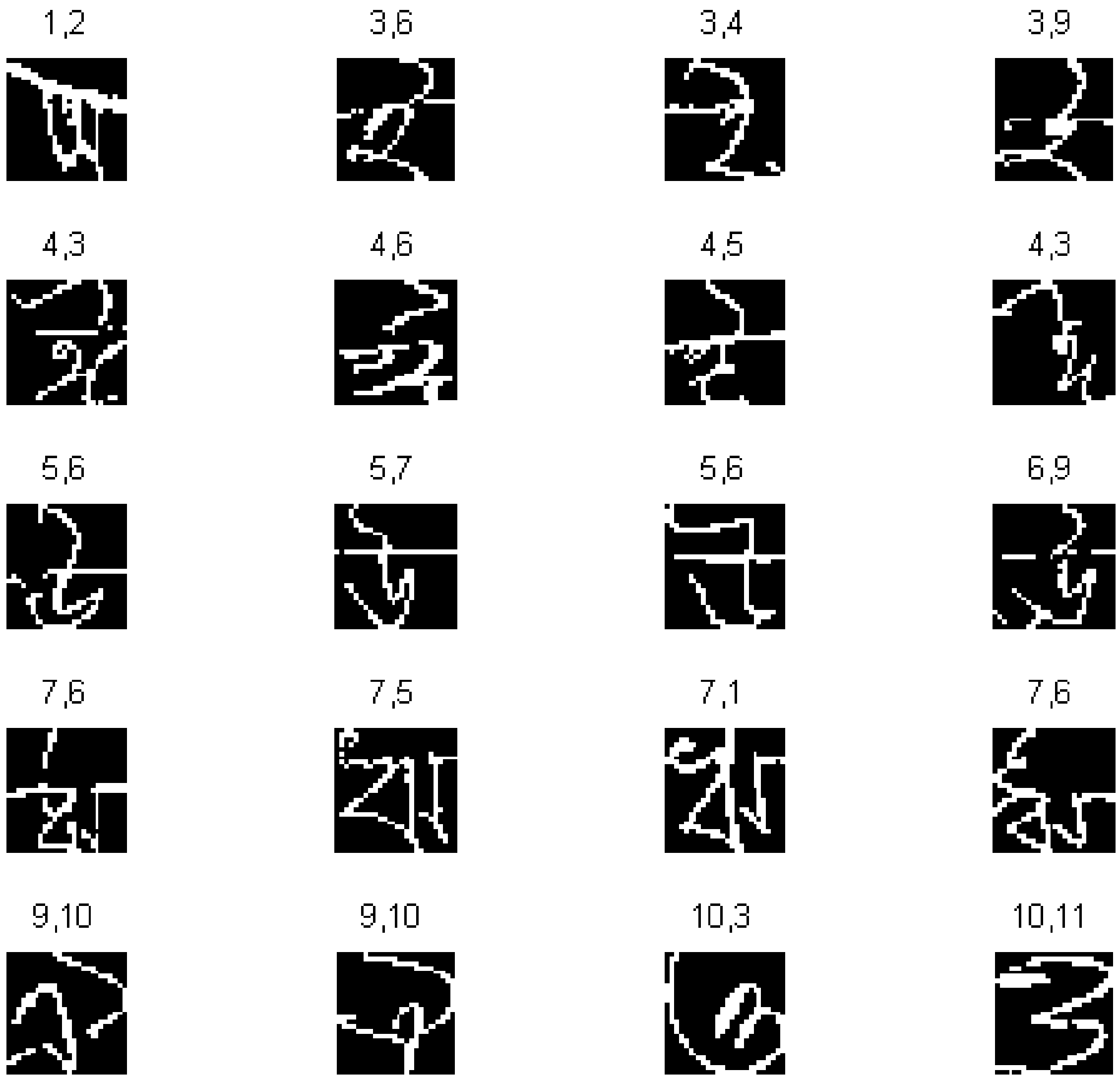


Figure 3.11: Misclassified characters (part 2)

Ch 4: Conclusion

We have used artificial neural network (ANN) for recognizing Bangla vowels. We collected the data ourselves and used ANN with back-propagation to classify the vowels. The best classification rate found by the system is 79.4% for 207 hidden layers with directional features obtained using a Gabor filter.

To recognize a pattern of hand writing is a complicated task though the process is simple. There are some observations of our thesis. Firstly, the outcomes depend on both good handwriting and a proper network. Secondly, a preprocessing method is very significant to get a good result. We scanned the image at 300 dpi resolution and normalized each character to 27×26 pixel.

It is also observed that, the possible changes of character size can differ the accuracy rate to identify an unknown character. In our case, after changing the amount of samples and character size the accuracy has been improved. We used Gabor filter for feature extraction. The third observation of us is that the feature extraction has some parameters. And those parameters are also responsible to accurate a neural network to identify a character with least error. But with feature extraction, we got 79.4% accuracy rate. Our most important observation is with the increase of samples, the accuracy rate is increased. A neural network is easier method to implement samples for identifying unknown characters. But the different handwriting is the basic obstacle to make more accurate implementation.

In the future this work could be extended to include the consonants. Also, more data samples can be collected.

Bibliography

[1] Anil K. Jain, Robert P.W. Duin, and Jianchang Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37, 2000.

[2] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, pp. 2271 - 2285, 2003.

[3] H. Fujisawa, "Forty years of research in character and document recognition-an industrial perspective," *Pattern Recognition,* vol. 41, no. 8, pp. 2435 - 2446, 2008.

[4] S. Mori, C.Y. Suen and K. Kamamoto, "Historical review of OCR research and development," in *Proceedings of IEEE*, 1992.

[5] Paul Gader, Brian Forester, Margaret Ganzberger, Andrew Gillies, Brian Mitchell, Michael Whalen, Todd Yocum, "Recognition of handwritten digits using template and model matching," *Pattern Recognition*, vol. 24, no. 5, pp. 421 - 423, 1991.

[6] Anil K. Jain, "Representation and Recognition of Handwritten Digits Using Deformable Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1386 - 1391, 1997.

[7] Luisa Mico, Jose Oncina, "Comparison of fast nearest neighbour classifiers for handwritten character recognition," *Pattern Recognition Letters,* vol. 19, pp. 351 - 356, 1998.

[8] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, Wiley, 2001.

[9] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.

[10] Simon Haykin, Neural Networks A Comprehensive Foundation, Prentice Hall, 1999.

[11] G. P. Zhang, "Neural Networks for Classification: A Survey," *IEEE Transactions on System, Man, and Cybernatics - Part C: Applications and Reviews,* vol. 30, no. 4, pp. 451 - 462, 2000.

[12] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[13] Yan Le Cun, et al., "Handwritten Character Recognition Using. Neural Network Architecturesee," in *Proceedings of the 4th USPS Advanced Technology Conference*, Washington D.C., 1990.

[14] Nibaran Das, Brindaban Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu,, "Handwritten Bangla Basic and Compound character recognition using MLP and SVM

classifier," *JOURNAL OF COMPUTING,* vol. 2, no. 2, pp. 109 - 115, 2010.

[15] Md. Azher Uddin, "Handwritten Bangla Character Recognition Using Artificial Neural Network," *IOSR Journal of Computer Engineering,* vol. 16, no. 3, pp. 33 - 38, 2014.

[16] Md. Alamgir Badsha Md. Akkas Ali Dr. Kaushik Deb Md. Nuruzzaman Bhuiyan, "Handwritten Bangla Character Recognition Using Neural Network," *nternational Journal of Advanced Research in Computer Science and Software Engineering,* vol. 2, no. 11, pp. 307 - 312, 2012.

[17] Seiichi Uchida, Hiroaki Sakoe, "A Survey of Elastic Matching Techniques for Handwritten Character Recognition," *IEICE - Transactions on Information and Systems,* Vols. E88 - D, no. 8, pp. 1781 - 1790, 2005.

[18] December 2015. [Online]. Available: https://en.wikipedia.org/wiki/Portable_Network_Graphics.

[19] December 2015. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm.

[20] December 2015. [Online]. Available: https://en.wikipedia.org/wiki/Otsu's_method.

[21] A. K. Jain, "Artificial neural networks: a tutorial," *IEEE Computer,* vol. 29, no. 3, pp. 31 - 44, 1996.

[22] D. Trier, A. K. Jain, T. Taxt,, " Feature Extraction Method for Character Recognition - A Survey," *Pattern Recognition,* vol. 29, no. 4, pp. 641 - 662, 1996.

[23] D. Gabor, "Theory of Communications," *Journal of the Institute of Electrical Engineers,* vol. 93, pp. 429-457, 1946.