Chapter 2. Complexities of Printed Gujarati Character Recognition and related work

2.1. Introduction

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories. It is natural that we should seek to design and build machines that can recognize patterns. Optical Character Recognition is a part of pattern recognition task which is used to convert the digital images of printed documents in to files of editable text by using computers. This chapter describes the development of the OCR for Gujarati script and related work in other Indian languages. The organization of the chapter is shown below:

This chapter is divided in to six sections. After giving introduction in the first section, we discuss Machine learning and pattern recognition tasks in the second section. Third and fourth sections discuss the details of OCR and complexities of Gujarati script for the development of OCR respectively. Status of work related to OCR technology for other Indian scripts is presented in the fifth section followed by a summary in the sixth section.

2.2. Machine Learning and Pattern Recognition

Machine learning is a branch of artificial intelligence, and is concerned with the design and development of algorithms and techniques that allow computers to "learn". The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. Therefore, machine learning is closely related to Statistics. Machine learning algorithms are routinely applied to pattern recognition tasks.

In the case of character recognition problems, good results can be obtained by adopting a machine learning approach in which a large set of *N* patterns of characters $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ called a training set is used to tune the parameters of an adaptive model. The categories of characters in the training set are known in advance, typically by inspecting them individually and hand-labeling the identity of the corresponding character.

The result of running the machine learning algorithm can be expressed as a function *Y(X)* which takes a new character image X as input and that generates an output function *Y*, encoded in the same way as the target vectors. The precise form of the function *Y(X)* is determined during the training data. Once the model is trained it can then determine the identity of new character images, which are said to comprise a *test set*. The ability to categorize correctly new examples that differ from those used for training is known as generalization.

For most practical applications, the original input variables are typically pre-pocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve. For instance, in the character recognition problem, the images of the characters are typically translated and scaled so that each character is contained within a box of a fixed size. This greatly reduces the variability within each character class, because the location and scale of all the characters are now the same, which makes it much easier for a subsequent pattern recognition algorithm to distinguish between the different classes. This pre-processing stage is sometimes also called feature extraction.

Pre-processing might also be performed in order to speed up computation. For example, if the goal is real-time face detection in a high-resolution video stream, the computer must handle huge numbers of pixels per second, and presenting these directly to a complex pattern recognition algorithm may be computationally infeasible. Instead, the aim is to find useful features that are fast to compute, and yet that also preserve useful discriminatory information enabling faces to be distinguished from non-faces. These features are then used as the inputs to the pattern recognition algorithm.

Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as supervised learning problems. Cases such as the character recognition example, in which the aim is to assign each problems. If the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the yield in a chemical manufacturing process in which the inputs consist of the concentrations of reactants, the temperature, and the pressure.

In other pattern recognition problems, the training data consists of a set of input vectors X without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

The classification task can broadly be performed by two ways viz. Statistical learning and Artificial Neural Networks. In the following subsection, we describe the pattern classification by Statistical learning approach (chapter 5) while Artificial Neural Networks approach is to be discussed in the chapter-4 and 6.

This section is divided into the three subsections. The first subsection introduces the approach of Statistical learning theory. The problem of pattern recognition using Statistical learning approach is discussed in second subsection and the third subsection correlates the use of Statistical learning in neural networks.

2.2.1 Statistical Learning

Suppose the pattern vector, **X**, is a random variable whose probability distribution for category 1 is different than it is for category 2. (While we present the results for two categories, these results can easily be extended to the case of multiple categories.) Assume that the two categories have the two probability distributions (perhaps

probability density functions), $p(\mathbf{X}|1)$ and $p(\mathbf{X}|2)$. Given a pattern, X, we want to use statistical techniques to determine its category-that is, to determine from which distribution it was drawn.

We shall make use of Bayes' theorem which is used to find out inverse conditional probability by using sum and product rules of probabilities. The sum and product rules of probabilities and Bayes' theorem can be briefly defined as follows:

1. The Rules of Probability:

   For the random variables X and Y, the sum rule and the product rule are defined as under:

   Sum Rule: $p(X) = \sum_{Y} p(X,Y)$

   Product Rule: $p(X,Y) = p(Y \mid X)p(X)$

   Where $p(X,Y)$ is a joint probability i.e. the probability of X and Y, the quantity $p(Y \mid X)$ is a conditional probability i.e. the probability of Y given X and the quantity $p(X)$ is a marginal probability or simply "the probability of X".

2. Bayes' theorem:

   From the product rule, together with the symmetry property $p(X,Y) = p(Y,X)$, we obtain the relationship between conditional probabilities as follows:

   $$p(X,Y) = p(Y,X)$$

   $$p(Y \mid X)p(X) = p(X \mid Y)p(Y)$$

   $$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}, \qquad \text{where } p(X) = \sum_{j} p(X \mid y_j)p(y_j)$$

   which is called Baye's theorem and which plays a central role in pattern recognition and machine learning.

Our problem is to find a decision rule against $P(y_j)$ that minimizes the expected loss [38] . A general decision rule is a function $\alpha(\mathbf{X})$ that tells us which action to take for every possible observation. To be more specific, for every **X** the decision function $\alpha(\mathbf{X})$ assumes one of the a values $\alpha_1, \alpha_2, \ldots, \alpha_a$.

In developing a decision rule, it is necessary to know the relative seriousness of the two kinds of mistakes that might be made. (We might decide that a pattern really in category 1 is in category 2, and vice versa.) . Loss function plays an important role in the development of a decision rule. The loss function states exactly how costly each action is, and is used to convert a probability determination into a decision. The function can be described as below:

We describe this information by a loss function $\lambda(i \mid j)$, for i,j=1,2. $\lambda(i \mid j)$ represents the loss incurred when we decide a pattern is in category i when really it is in category j. We assume here that $\lambda(1 \mid 1)$ and $\lambda(2 \mid 2)$ are both 0. For any given pattern, **X**, we want to decide its category in such a way that minimizes the expected value of this loss.

Given a pattern, **X**, if we decide category i, the expected value of the loss will be:

$$L_X(i) = \lambda(i \mid 1)p(1 \mid \mathbf{X}) + \lambda(i \mid 2)p(2 \mid \mathbf{X})$$

Where $p(j \mid \mathbf{X})$ is the probability that given a pattern **X**, its category is j. Our *decision rule* will be to decide that **X** belongs to category 1 if $L_{\mathbf{X}}(1) \leq L_{\mathbf{X}}(2)$, and to decide on category 2 otherwise.

We can use Bayes' Rule to get expressions for $p(j \mid \mathbf{X})$ in terms of $p(\mathbf{X} \mid j)$, which we assume to be known (or estimatible):

$$p(j \mid \mathbf{X}) = \frac{p(\mathbf{X} \mid j)p(j)}{p(\mathbf{X})}$$

Where $p(j)$ is the (a priori) probability of category j (one category may be much more probable than the other); and $p(\mathbf{X})$ is the (a priori) probability of pattern **X** being the

pattern we are asked to classify. Performing the substitutions given be Bayes' Rule, our decision rule becomes:

$$\lambda(1|1)\frac{p(\mathbf{X}|1)p(1)}{p(\mathbf{X})}+\lambda(1|2)\frac{p(\mathbf{X}|2)p(2)}{p(\mathbf{X})}$$

$$\lambda(2|1)\frac{p(\mathbf{X}|1)p(1)}{p(\mathbf{X})}+\lambda(2|2)\frac{p(\mathbf{X}|2)p(2)}{p(\mathbf{X})}$$

Using the fact that $\lambda(i \mid i) = 0$, and noticing that $p(X)$ is common to both expressions, we obtain,

Decide category 1 iff:

$$\lambda(1|2)p(\mathbf{X}|2)p(2) \le \lambda(2|1)p(\mathbf{X}|1)p(1)$$

If $\lambda(1|2) = \lambda(2|1)$ and if $p(1) = p(2)$, then the decision becomes particularly simple:

Decide category 1 iff:

$$p(\mathbf{X}|2) \le p(\mathbf{X}|1)$$

Since $p(\mathbf{X}|j)$ is called the likelihood of j with respect to **X**, this simple decision rule implements what is called a maximum-likelihood decision.

For multi category problems Gaussian (or normal) distributions or Conditionally Independent Binary Components may be used. The use of statistical learning theory in the pattern recognition problem is to be discussed in the following subsection.

2.2.2 Statistical Learning and Pattern recognition

In the practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find the patterns in data.

In this subsection, we discuss the statistical characterization of neural networks by describing a learning theory that addresses the fundamental issue of how to control the generalization ability of a neural network in mathematical terms. A model of supervised learning consists of three interrelated components abstracted in mathematical terms as follows (Vapnik, 1992, 1998):

1. Environment. The environment is stationary, supplying a vector **X** with a fixed but unknown probability distribution function $F_{\mathbf{X}}(\mathbf{X})$.
2. Teacher. The teacher provides a desired response **d** for every input vector **X** received from the environment, in accordance with a conditional probability distribution $F_{\mathbf{X}}(\mathbf{X}|d)$ that is also fixed but unknown. The desired response d and input vector X are related by

$$\mathbf{d} = f(\mathbf{X}, v) \tag{2.1}$$

where v is a noise term, permitting the teacher to be "noisy".

3. Learning machine (algorithm). The learning machine (neural network) is capable of implementing a set of input-output mapping functions described by

$$\mathbf{y} = F(\mathbf{X}, \mathbf{w}) \tag{2.2}$$

where **y** is the actual response produced by the learning machine in response to the input **X**, and **w** is a set of free parameters (synaptic weights).

Equations (2.1) and (2.2) are written in terms of the examples used to perform the training.

The supervised learning problem is that of selecting the particular function $F(\mathbf{X}, \mathbf{w})$ that approximates the desired response **d** in an optimum fashion, with "optimum" being defined in some statistical sense [18]. In probability theory, a sequence or other collection of random variables is independent and identically distributed (i.i.d.) if each has the same probability distribution as the others and all are mutually independent. The selection itself is based on the set of N independent, identically distributed training examples given as $\varsigma = \{(\mathbf{X}_i, d_i)\}_{i=1}^{N}$. In a probability theory, a sequence or other collection of random variables is independent, identically distributed if each has the same probability distribution as the others and all are mutually independent.

Each example pair is drawn by the learning machine from ς with a joint probability distribution function $F_{\mathbf{X},\mathbf{D}}(\mathbf{X}, \mathbf{d})$, which, like the other distribution functions, is also fixed but unknown.

2.2.3 Statistical Nature of the Learning processes of Neural Networks

From a computational point of view, the supervised learning (or training) of a neural network is visualized as process of adaptation or evolution of the weight matrix **w** in such a way that the computed outputs of the network are driven towards the desired output for the input vectors of the training data.

Viewing the training process from the statistical perspective provides a deeper understanding of the training process and the training algorithms. From the statistical perspective, the focus is on the deviation between a "target" function $f(\mathbf{X})$ and the actual function $F(\mathbf{X},\mathbf{w})$ realized by the neural network. The deviation is expressed in statistical terms. This corresponds to confining the attention to a single output neuron but there is no loss of generality in this.

Consider a random vector **X** consisting of a set of independent variables and a random scalar D representing the dependent variable. Let there be N realizations of the random vector **X** denoted by $\{\mathbf{x}_i\}_{i=1}^{N}$ and a corresponding set of random scalar D denoted by $\{d_i\}_{i=1}^{N}$. These realizations constitute the training set denoted by

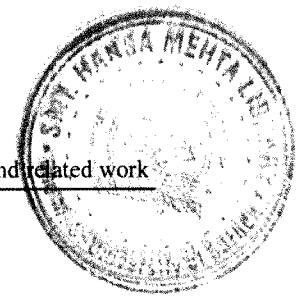
$$\mathrm{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}. \qquad \text{(i)}$$

Specifying a neural network model for training corresponding to proposing a model of type

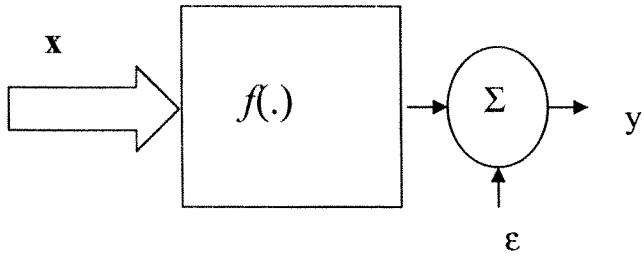
$$D = f(\mathbf{X}) + \varepsilon \qquad \text{(ii)}$$

where $f(.)$ is a deterministic function of **X** and ε is a random expectational error.

The statistical model specified in figure (2.1) is called a regressive model. ε in general is a random variable with zero mean and positive probability of occurrence.

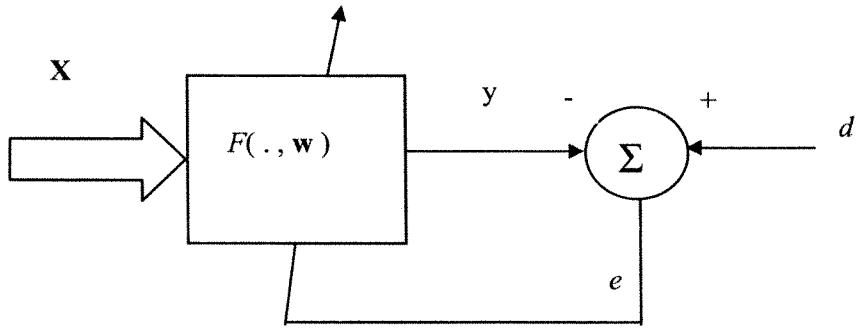


This model can be diagrammatically represented as:



[Fig. 2.1 Regressive model of *f(.)*]

This model is a mathematical description of a stochastic environment. Its purpose is to use vector **X** to predict the dependent variable *D*. The corresponding neural network model is depicted in the following diagram:



[Fig. 2.2 Neural network model]

The purpose of this physical neural network model is to encode the empirical knowledge represented by the training sample set T in to a corresponding set of synaptic weight vector **w**, which can be indicated by

$$T \longrightarrow \mathbf{w}$$

The neural network provides an approximation to the regressive model.

Let the random variable *Y* represent the actual response of the neural network to the input vector **x**, i.e.

$$Y = f(\mathbf{x},\mathbf{w})$$

where $f(\ .\ ,\mathbf{w})$ is the input-output function realized by the neural network. Given the training data T, the weight vector w is obtained by minimizing the cost function

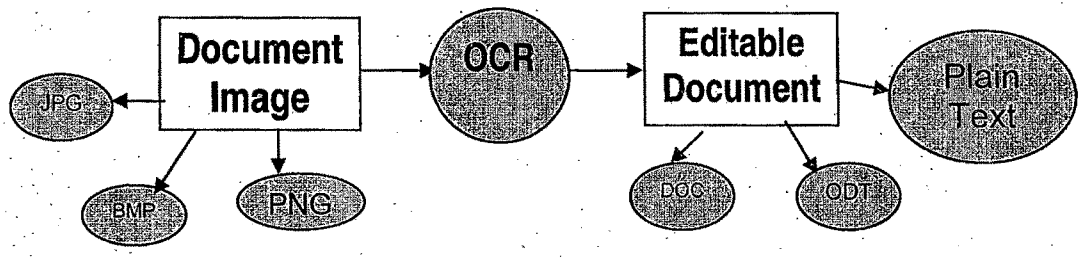
$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i, \mathbf{w})]^2$$

A training algorithm of the neural network is designed to minimize $E(\mathbf{w})$.

Artificial neural network architectures presented in the chapter 5 and 6 of the thesis are based on statistical learning theory.

2.3. Optical Character Recognition

Optical Character Recognition (OCR) is a typical application of machine learning and pattern recognition task. A Pictorial representation of the OCR problem is shown in the following block diagram.



[Fig. 2.3 Optical Character Recognition]

As shown in figure 2.3, various image files like .jpg, .bmp, .png etc are given as an input to OCR system. The output expected from the system is an editable file in a format like .doc, .odc, .txt etc.

Typically, the following five steps are to be performed in order to develop OCR system for any script.

1. Gray level scanning at an appropriate resolution, typically 300-1000 dots per inch (dpi).

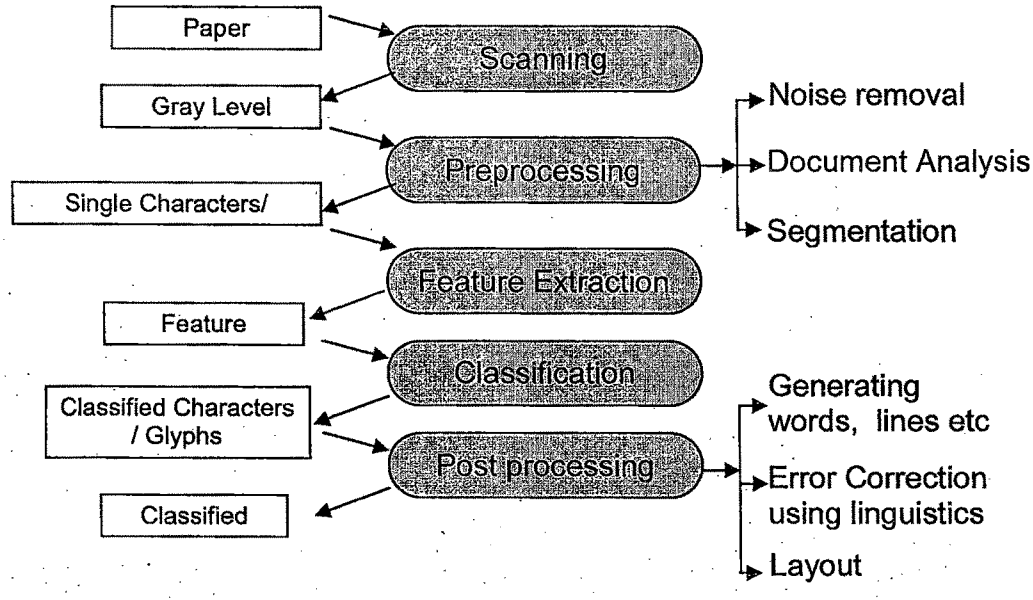
2. Preprocessing:
   {a} Binarization (2-level thresholding), using a global or a locally adaptive method
   {b} Segmentation to isolate individual characters
   {c} (Optional) conversion to another character recognition (eg. Skeleton or contour curve)

3. Feature extraction: It can be done by various Statistical or Mathematical like Statistical moments, Discrete Cosine Transform, Wavelets etc.

4. Recognition using one or more classifiers: Classification accuracy of each individual glyph is based on how useful the extracted features are. For this purpose various classification techniques are available like Nearest Neighbor Classifier (NNC), Artificial Neural Networks etc.

5. Contextual verification or post-processing: After the recognition of isolated glyphs, they are to be arranged at an appropriate place from where they were originally kept. There are some confusing characters, which fail to be recognized, will form a confusion set and ultimately affect the recognition accuracy. This set is to be tackled by sub-glyph level analysis at the time of post-processing.

The block diagram of the whole process is shown below:



[Fig. 2.4 Process of OCR]

2.4. Details of Gujarati Script and its complexities

Gujarati is also the name of the script used to write the Gujarati language spoken by about 50 million people in western India and many others located in almost all parts of the world. The Gujarati script has 11 vowels and 34+2[#] consonants. Apart from these basic symbols, other symbols used in the Gujarati script include a large number of (more than 250) consonant clusters called conjuncts and vowel modifiers (or dependent vowels) used to denote the attachment of vowels with the core consonants and conjuncts. The shapes of some Gujarati characters are similar to those of the phonetically corresponding characters of Devanagari script. As in the case of other Indic scripts, Gujarati also does not have the distinction of Lower and Upper Cases.

[#] Two conjuncts /ksha/ and /jya/ are treated as if they are basic consonant in Gujarati script

In spite of these similarities with the Devanagari script, Gujarati script has many distinct characteristics such as the absence of the so-called shirorekha (header line) in the script and differences in the shapes of many of the consonants and vowels etc.

Table 1a. Gujarati Script

Consonants
ક ખ ગ ઘ ઙ ચ છ જ ઝ ઞ ટ ઠ ડ ઢ ણ ત થ દ ધ ન પ ફ બ
ભ મ ય ર લ વ શ ષ સ હ ળ ક્ષ જ્ઞ
Vowels
અ આ ઇ ઈ ઉ ઊ એ ઍ ઐ ઓ ઑ ઔ અં
Some Conjuncts
ક્ક ચ્ચ ક્લ ચ્છ દ્ર ત્ર સ્પ સ્ત ર્વ ક્ક ત્ય ત્ર ત્ત દ્ય
Consonant – vowel
જા ગી જી બુ ખૂ હે કુ કૈ પૌ ડો કી કં
Conjunct –vowel
જ્ઞા કૈ ક્લે ચ્છિ દ્રી ત્રુ સ્ત્રૂ
Vowels modifiers
ા િ ી ુ ૂ ૃ ે ૈ ો ૌ ્

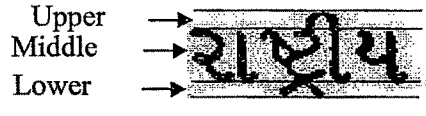
Table 1b. Conjunct Formation

1. Conjunct formation by connecting a part of preceding consonant to the following one
જ્ + ઞ = જ્ઞ, ક્ + લ = ક્લ, સ્ + પ = સ્પ
2. Conjunct taking completley new shape
દ્ + વ = દ્વ, ક્ + ર = ક્ર, ત્ + ર = ત્ર
3. Conjunct formation by Addition of some mark
દ્ + વ = દ્વ, ક્ + ર = ક્ર, ત્ + ર = ત્ર, ર્ + વ = ર્વ, દ્ + ર = દ્ર

The combinations of consonants and conjuncts with vowel modifiers occur very often in most of the Indic languages including Gujarati. The presence of the dependent vowel is denoted by attaching a symbol, unique for each vowel, to the consonant or conjunct, called a Dependent Vowel Modifier or Matra. The matra can appear before, after, above or below the core consonant or conjunct. Table 1a gives examples of Gujarati consonants, vowels, matras, other symbols and some of the conjuncts.

Apart from basic consonants, most of the Indian scripts, including Gujarati, often use consonant clusters (conjuncts). Here more than one consonants are combined together and all of them except the last one will be dead consonants (without inherent /a/). There are three possibilities for the shapes of the resulting conjuncts : 1. Conjunct shape is derived by connecting a part of preceding consonant to the following one 2. the conjunct takes a shape completely different from the shapes of the constituent consonants, 3. addition of some mark indicating conjunct formation in upper/middle/lower zone (mainly conjuncts involving /r/). Moreover, conjuncts may themselves occur in half forms. Table 1b shows some examples of conjunct formation and the resultant shapes.

Similar to the text written in Devanagari or Bangla scripts, text in Gujarati script can also be divided into three logical zones – Upper, Middle and Lower as shown in figure. 2.5. In [6], Jignesh Dholakia et al. have given an algorithm on zone separation for Gujarati script.



[Fig. 2.5. Zone separation]

2.5. Review of related work

Here we cite recent attempts by various groups of researchers to exploit the feature capturing capacities of wavelets for use in the recognition of symbols of various Indic scripts. Most of these attempts also have used the wavelet features in conjunction with various types of neural network architectures as classifiers. Radial Basis Function (RBF) Networks and dimensionality reduced features were used to recognize a subset of Kannada script from South India [7, 16, 17] with an accuracy approaching 95% by using wavelet features. A very innovative use of wavelet feature extraction is presented by Pujari et al [8], using Battle-Lemarie and Adelson-Simoncelli coefficients where the 32x32 input image is reduced to a 64 bit signature and recognized using a Hopfield based Dynamic Neural Network (DNN). The best performance reported in that paper is 93%. Performance of Daubechies D4 wavelets as feature extractor with the classic Multilayer Perceptron (MLP) classifier for online and offline handwritten character recognition of various Indic scripts was presented in [9, 10]. Some recent work on two different subsets of Gujarati script is given in [3] and [4]. The complexities of Gujarati script and related work will be described in the chapter 2 of the thesis. The concept of a Fuzzy Neuron is nicely introduced for Vietnamese Character Recognition problem by Bac Hoai Le et al [33]. In this paper the Self-Organizing Learning Algorithm of the Fuzzy neural networks is introduced and achieved significantly good recognition rate. The neural network base nonparametric density estimation namely Generalized Regression Neural Networks (GRNN) is used for Arabic isolated word recognition problem by Abderranhame

Amrouche et. al [34] and the results are compared with traditional MLP. A wonderful review material is found based on Wavelet Networks, Wavenets, Fuzzy Wavenets and their application for the classification problems [35]. Neural network based Radial Basis Function networks (RBFNs) and the innovative idea of subspace projection approach have been employed to recognize printed Kannada characters by using Haar wavelets and structural features [36]. And they have achieved 99.1% of accuracy by subspace projection approach.

2.6. Summary

The problem of development of OCR for Gujarati script is a complex task due to the absence of Shiro rekha and large number of conjuncts available in various combinations with matras, consonants and modifiers. It is observed from section 2.5 that the problem of OCR is characterized in to two major parts viz. one is feature extraction and the other is character recognition. As such various approaches for feature extraction and character recognition have been adopted as shown in the above section, but the use of wavelets as a feature extractor and ANN as a classifier are found to be the most prominent for the development of Gujarati script OCR. Development of Gujarati script OCR using these methods are to be discussed in the chapter 4 and chapter 6.