Handwritten Devnagri Script Recognition: A Review

Prashant M. Kakde¹, Dr. Sanjay M. Gulhane²

¹Research Scholar, S.G.B.A.U Amravati, India
²Professor, J.D.I.E.T, Yavatmal, India

Abstract—Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans’ job easier. Character Recognition (CR) has been an active area of research and due to its diverse applicable environment; it continues to be a challenging research topic. As the size of the vocabulary increases, the complexity of algorithms also increases linearly due to the need for a larger search space. The problem arise in Devnagari script character recognition using Zernike moments, fuzzy rule and quadratic classifier provide less accuracy and less efficiency. Classification methods based on learning from examples have been widely applied to character recognition from the 1990s and have brought forth significant improvements of recognition accuracies. This class of methods includes statistical methods, artificial neural networks, support vector machines (SVM), genetic algorithm, particle swarm optimization (PSO) algorithm etc. This includes comparative analysis using neural network, genetic algorithm, SVM and POS. This paper describes all these methods.

Index Terms— Handwritten Character Recognition, Genetic Algorithms, artificial neural networks, support vector machines (SVM), genetic algorithm, particle swarm optimization (PSO) algorithm

I. INTRODUCTION

Over the years, computerization has taken over large number of manual operations, one such example is offline handwritten character recognition, which is the ability of a computer system to receive and interpret handwriting input present in the form of scanned images. In the early stage of OCR (optical character recognition) development, template matching based recognition techniques were used [1]. The templates or prototypes in these early methods were designed artificially, selected or averaged from few samples. As the number of samples increased, these simple design methodologies, became insufficient to accommodate the shape variability of samples, and so, are not able to yield high recognition accuracies. To take full advantage of large volume of sample data, the character recognition community has turned attention to classification methods based on learning from examples strategy, especially based on artificial neural networks (ANNs) from the late 1980s and the 1990s. New learning methods, using support vector machines (SVMs), are now actively studied and applied in pattern recognition problems. Learning methods have beneficitated character recognition methods tremendously. They relieve us from painful job of template selection and tuning, and the recognition accuracies get improved significantly because of learning from large sample data.

Some excellent results have been reported [2][3][4]. Despite the improvements, the problem is far from being solved. The recognition accuracies of either machine-printed characters on degraded document image or freely handwritten characters are still insufficient.

Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed “off line” from a piece of paper by optical scanning (optical character recognition) or intelligent character recognition. Alternatively, the movements of the pen tip may be sensed “on line”, for example by a pen based computer screen surface.

Handwritten character recognition, irrespective of the script, finds potential application areas for automation in various fields like postal automation [5] [6], bank automation [7] [8], form filling etc. Handwritten character recognition for Indian scripts [9] is quite a challenging task for the researchers. This is due to the various characteristics of these scripts like their large character set, complex shape, presence of modifiers, presence of compound characters and similarity between characters. Marathi script derived from Devanagari, is an official language of Maharashtra. It is the 4th most spoken language in India and 15th most spoken language in the world. Marathi script consists of 16 vowels and 36 consonants making 52 alphabets. Marathi is written from left to right. It has no upper and lower case characters. Every character has a horizontal line at the top called as the header line. The header line joins the characters in a word. Vowels are combined with consonants with the help of specific characteristic marks. These marks occur in line, at the top, or at the bottom of a character in a word. Marathi also has a complex system of compound characters in which two or more consonants are combined forming a new special symbol. Compound characters in Marathi script occur more frequently in the script as compared to other languages derived from Devanagari. The occurrence of compound characters in Marathi is found to be about 11 to 12% whereas in other scripts of Devanagari and Bangla script, it is just 5 to 7% [10]. The compound characters exhibit following features: the consonants in the compound character are not joined in an arbitrary manner but the combination of some specific characters is done in order to give a meaningful combination.
II. CHARACTERISTICS OF DEVNAGARI SCRIPT

Devnagari script is different from Roman script in several ways. This script has two-dimensional compositions of symbols: core characters in the middle strip, optional modifiers above and/or below core characters. Two characters may be in shadow of each other. While line segments (strokes) are the predominant features for English, most of the characters in Devnagari script is formed by curves, holes, and also strokes. In Devnagari language scripts, the concept of upper-case, the lower-case characters is absent. It consists of 14 vowels and 33 consonants. Vowels occur either in isolation or in combination with consonants. Apart from vowels and consonants characters called basic characters, there are compound characters in Devnagari script, which are formed by combining two or more basic characters. Coupled to this in Devnagari script there is a practice of having twelve forms of modifiers with each for 33 consonants, giving rise to modified shapes which, depending on whether the modifier is placed to the left, right, top or bottom of the character. The net result is that there are several thousand different shapes or patterns, which makes Devnagari OCR more difficult to develop.

![Fig.1. Printed samples of Devnagari characters](image)

(a) Vowels (b) Consonants.

III. FEATURE EXTRACTION

In this section we give a brief description of the feature sets used in our proposed multiple classifier system. Chain code histogram features are extracted by chain coding the contour points of the scaled character bitmapped image.

A. Chain Code Histogram of Character Contour

Given a scaled binary image, we first find the contour points of the character image. We consider a 3 x 3 window surrounded by the object points of the image. If any of the 4-connected neighbor points is a background point then the object point (P), as shown in Fig.2 is considered as contour point.

![Fig 2. Contour point detection](image)

The contour following procedure generates a contour representation called “chain coding” that is used for contour following proposed by Freeman, shown in Fig.3a.

![Fig 3. Chain Coding: (a) direction of connectivity, (b) 4-way connectivity, (c) 8-connectivity. Generate the chain code by detecting the direction of the next-in-line pixel](image)

The chain code for the character contour will yield a smooth, unbroken curve as it grows along the perimeter of the character and completely encompasses the character. When there is multiple connectivity in the character, then there can be multiple chain codes to represent the contour of the character. We chose to move with minimum chain code number first. We divide the contour image in 5 x 5 equal sub-images. In each of these sub-images, the frequency of the 8-way direction code is computed and a histogram of chain codes is prepared for each block. Thus we get 5 x 5 x 8 = 200 chain code features for recognition.

B. Moment based features

Region moment representations interpret a normalized gray-level image function as a probability density of a 2D random variable. Assuming that non-zero pixel values represent regions, moments can be used for binary or gray-level transformations. Translation invariance can be achieved by using the central moments. For a digital image the central moments can be expressed as:-
In each generation, the fitness of the entire population is evaluated, and multiple individuals are selected from the present population based on their fitness. These are modified, mutated, or recombined to make a new population, which becomes present in the next iteration of the algorithm. Usually, the solutions are represented in strings of 0s and 1s, though different encodings are also possible. So, evolutionary algorithms play on populations, in its place of coming to one solution.

When the features of the characters in the sub-word are determined, the next phase is to recognize the characters of the sub-word. The genetic algorithm approach will be used for this purpose.

**B. Artificial Neural Networks**

In Artificial Neural Networks, including multilayer perceptron (MLP), radial basis function (RBF) network [12], the probabilistic neural network (PPN) [13], higher-order neural network (HONN) [14], etc., have been widely applied to pattern recognition. The connecting weights are usually adjusted to minimize the squared error on training samples in supervised learning. Using a modular network for each class was shown to improve the classification accuracy [15]. A network using local connection and shared weights, called convolutional neural network, has reported great success in character recognition [16]. Kumar and Singh [15] proposed a Zernike moment feature based approach for Devnagari handwritten character recognition. They used an artificial neural network for classification. Bhattacharya et al [18] [19] proposed a Multi-Layer Perceptron (MLP) neural network based classification approach for the recognition of Devnagari handwritten numerals. S.Arora [20] proposed a MLP designed on some statistical features for handwritten devnagari characters recognition. The RBF network can yield competitive accuracy with the MLP when training all parameters by error minimization [21]. The HONN is also called as functional-link network, polynomial network or polynomial classifier (PC). Its complexity can be reduced by dimensionality reduction before polynomial expansion [22] or polynomial term selection [23]. Vector quantization (VQ) networks and auto association networks, with the sub-net of each class trained independently in unsupervised learning, are also useful for classification. The learning vector quantization (LVQ) of Kohonen [24] is a supervised learning method and can give higher classification accuracy than VQ. Some improvements of LVQ learn prototypes by error minimization instead of heuristic adjustment [25].

Another method in Neural Network is feed method. A Neural Network consists of two basic kinds of elements, neurons and connections. Neurons connect with each other through connections to form a network. This is a simplified theory model of the human brain.
A Neural Network often has multiple layers and neurons of a certain layer connect neurons of the next level in some way. Every connection between them is assigned with a weight value.

At the beginning, input data are fed into the neurons of the first layer, and by computing the weighted sum of all connected first layer neurons, we can get the neuron value of a second layer neuron and so on finally, we can reach the last layer, which is the output. All the computations involved in operating a Neural Network are a bunch of dot products. We need to train our network with sample inputs and compare the outcomes with our desired answers. Some algorithm can take the errors as inputs and modify the network [5].

Neural network is first trained by multiple sample images of each alphabet. Then in recognition process, the input image is directly given to neural network and recognized symbol is outputted. Although even best of the recognition approaches are able to recognize all those texts that human can. Major reason for this has been wide variation in writing practices and styles of different people and lot of context based information that human brain uses to interpret any text sample.

### C. Support Vector Machines

The objective of any machine capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data. The principle of an SVM is to map the input data onto a higher dimensional feature space nonlinearly related to the input space and determine a separating hyperplane with maximum margin between the two classes in the feature space[22]. A support vector machine is a maximal margin hyperplane in feature space built by using a kernel function in gene space. This results in a nonlinear boundary in the input space. The optimal separating hyperplane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space. Commonly used kernels include:-

1. Linear Kernel:
   \[ K(x, y) = x \cdot y \]

2. Radial Basis Function (Gaussian) Kernel :
   \[ K(x, y) = \exp \left( \frac{-||x - y||^2}{2\sigma^2} \right) \]

3. Polynomial Kernel:
   \[ K(x, y) = (x \cdot y + 1)^d \]

An SVM in its elementary form can be used for binary classification. It may, however, be extended to multiclass problems using the one-against-the-rest approach or by using the one-against-one approach. We begin our experiment with SVM’s that use the Linear Kernel because they are simple and can be computed quickly. There are no kernel parameter choices needed to create a linear SVM, but it is necessary to choose a value for the soft margin (C) in advance. Then, given training data with feature vectors \( x_i \) assigned to class \( y_i \in \{-1,1\} \) for \( i = 1,\ldots,l \), the support vector machines solve

\[ \text{Subject to } y_i (K(\omega, x_i) + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]

Where \( \xi \) is an l-dimensional vector, and \( \omega \) is a vector in the same feature space as the \( x_i \). The values \( \omega \) and \( b \) determine a hyper plane in the original feature space, giving a linear classifier. A priori, one does not know which value of soft margin will yield the classifier with the best generalization ability. We optimize this choice for best performance on the selection portion of our data.

### D. Particle Swarm Optimization

The Particle Swarm Optimization algorithm design by Kennedy and Eberhart in 1995. PFO is a population-based searching method which imitates the social behavior of bird flocks or fish schools.
In PSO population and the individuals are called a “swarm” and “particles”, respectively. Each particle moves in the swarm with a velocity that is adjusted according to its own flying experience and retains the best position it has ever encountered in memory. The best local and global positions ever encountered by all particles of the swarm are also communicated to all other particles. The popular form of particle swarm optimizer is defined in the following equations and in the flow chart in Figure 5 [11][13].

\[ V_{id}(t+1) = W \cdot V_{id}(t) + C_1 \cdot R_1 \cdot (P_{id}(t) - X_{id}(t)) + C_2 \cdot R_2 \cdot (P_{g}(t) - X_{id}(t)) \]  

(1)

\[ X_{id}(t + 1) = X_{id}(t) + V_{id}(t + 1) \]  

(2)

Where:

- \( V_{id} \): is the velocity of particle \( i \) along dimension \( d \).
- \( X_{id} \): is the position of particle \( i \) in dimension \( d \).
- \( C_1 \): is a weight applied to the cognitive learning portion.
- \( C_2 \): is a similar weight applied to the influence of the social learning portion.
- \( R_1, R_2 \): are separately generated random numbers in the range of zero and one.
- \( W \): is the inertia weight.

V. PERFORMANCE EVALUATION

The performance evaluation shows the the result of different methods for handwritten devnagari characters recognition. As till date no standard dataset is available for Handwritten Devnagari Characters we collected some samples from ISI. Table shows the compression between the SVM and neural network method accuracy on ISI dataset.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Test set</th>
<th>Training set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>80.67%</td>
<td>94.77%</td>
</tr>
<tr>
<td>Multiple Neural Network</td>
<td>Min</td>
<td>60.92%</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>74.65%</td>
</tr>
<tr>
<td>Classifier Combination</td>
<td></td>
<td>84.12%</td>
</tr>
<tr>
<td>(ANNs)</td>
<td>Weighted majority</td>
<td>70.38%</td>
</tr>
<tr>
<td></td>
<td>(top1)</td>
<td>90.74% (top5)</td>
</tr>
<tr>
<td></td>
<td>93.31% (top5)</td>
<td>93.31%</td>
</tr>
</tbody>
</table>

Table 1. Results of SVM and ANN on ISI dataset

Table 2 shows the different results for the all the above methods which used by given authors in there paper for the handwritten devnagari characters recognition. It also shows the accuracy of all the methods for character reorganization.

<table>
<thead>
<tr>
<th>SR. NO</th>
<th>Proposed By</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kumar and Singh [24]</td>
<td>SVM</td>
<td>80%</td>
</tr>
<tr>
<td>3</td>
<td>M. Hanmandlu, O.V. R. Murthy, V.K. Madasu[26]</td>
<td>Max</td>
<td>74.65%</td>
</tr>
<tr>
<td>4</td>
<td>S. Arora, D. Bhattacharjee, M. Nasipuri, [18]</td>
<td>SVM</td>
<td>90.74% (top5)</td>
</tr>
<tr>
<td>5</td>
<td>U. Pal, N. Sharma, T. Wakabayashi and F. Kimura [27]</td>
<td>SVM</td>
<td>93.31% (top5)</td>
</tr>
</tbody>
</table>

Table 2. Comparison of Results

<table>
<thead>
<tr>
<th>SR. NO</th>
<th>Proposed By</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sushama Shelke [28]</td>
<td>Neural Network</td>
<td>94.22 %</td>
</tr>
<tr>
<td>2</td>
<td>Vedgupt Saraf [29]</td>
<td>Genetic Algorithm</td>
<td>97.68%</td>
</tr>
<tr>
<td>3</td>
<td>Sandip Arora [18]</td>
<td>SVM</td>
<td>94.77%</td>
</tr>
<tr>
<td>4</td>
<td>C. Namrata Mahender[30]</td>
<td>PSO</td>
<td>80.78%</td>
</tr>
</tbody>
</table>

Table 3. Comparison of Results for different methods
VI. CONCLUSION

This paper has attempted to review a significant number of papers to cover the recent development in the field of handwritten devnagari characters recognition. Present study reveals that various methods have been used. It includes a Genetic Algorithms, artificial neural networks, support vector machines (SVM), genetic algorithm, particle swarm optimization (PSO) algorithm and many other. The list of references to provide more detailed understanding of the approaches described is enlisted. We apologize to researchers whose important contributions may have been overlooked.

REFERENCES


