ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

Survey of Methods for Character Recognition

Suruchi G. Dedgaonkar, Anjali A. Chandavale, Ashok M. Sapkal

Abstract*— Character recognition has long been a critical area of the Artificial Intelligence. Recognition is a trivial task for humans, but to make a computer program that does character recognition is extremely difficult. Recognizing patterns is just one of those things humans do well and computers don't. The reasons for this are the many sources of variability, abstraction and absence of hard-and-fast rules that define the appearance of a visual character. Hence rules need to be heuristically deduced from samples. This paper provides a review for various available methods. Character recognition methods are listed under two main headlines. The "Offline" methods use the static image properties. The offline methods are further divided into four methods, which are Clustering, Feature Extraction, Pattern Matching and Artificial Neural Network. The online methods are subdivided into k-NN classifier and direction based algorithm. Thus, an appreciation is provided for the range of techniques available for character recognition. The methods are discussed in detail throughout the paper.

***Index Terms*— Character Recognition, Feature Extraction, Clustering, Pattern Matching, Neural Network, ANN, OCR.**

I. INTRODUCTION

Character recognition is the process to classify the input character according to the predefined character class. With the increasing interest of computer applications, modern society needs that the computer should read the text. The text may be in the form of scanned handwritten document or typed text in various fonts or a combination of both. The character recognition system helps in making the communication between a human and a computer easy. Classical methods in recognition are not perfect for the recognition of visual characters due to the following reasons [11]:

1. The 'same' characters differ in sizes, shapes and styles from person to person and even from time to time with the same person. The source of confusion is the high level of abstraction: there are thousands styles of type in common use and a character recognition program must recognize most of these.

2. Like any image, visual characters are subject to spoilage due to noise. Noise consists of random changes to a pattern, particularly near the edges. A character with much noise may be interpreted as a completely different character by a computer program.

3. There are no hard-and-fast rules that define the appearance of a visual character. Hence rules need to be heuristically deduced from the samples. Character recognition system is useful in license plate recognition system, smart card processing system, automatic data entry, bank cheque /DD processing, money counting machine, postal automation, address and zip code recognition, writer identification etc.

There exist several different techniques for recognizing characters. One distinguishes characters by the number of loops in a character and the other by direction of their concavities. These methods can be used one after the other to increase accuracy and speed for recognition.

II. CHARACTER RECOGNITION TECHNIQUES

Some approaches take a holistic approach, recognizing entire words, while others focus more on recognizing individual characters. Holistic approaches incur more computational cost since there are more models, but have more expressive and discriminative power since the visual cues are gathered over large areas. Fig. 1 shows the classification of character recognition techniques. Basically the character recognition can be done using online and offline methods. These methods are discussed throughout the paper.

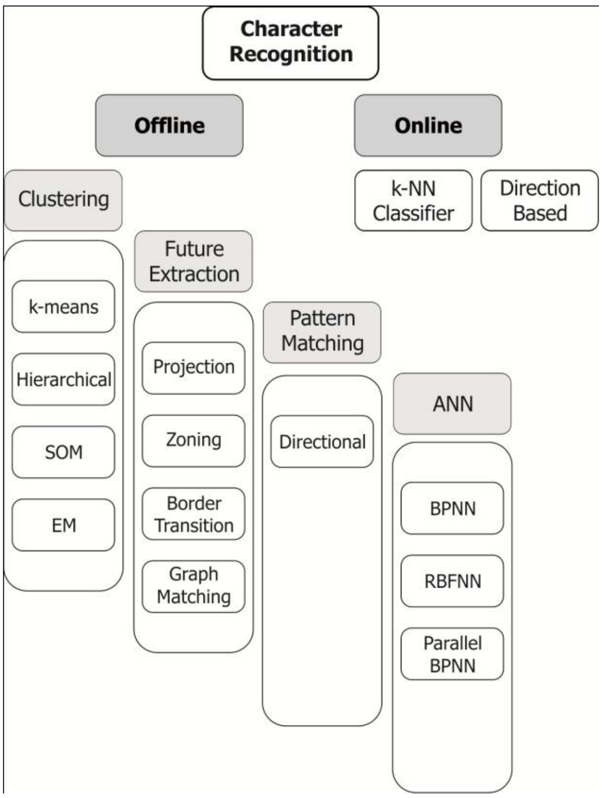

Fig. 1 Types of Character Recognition Techniques

A. On-line recognition

Online handwriting recognition has gained interest due to increase in usage of hand held devices. Nonparametric methods have recognition time proportionate to the training set size. These methods use all points per stroke for calculating the similarity measurement. The incorporation of keyboard being difficult in the hand held devices demands for alternatives, and in this respect, online method of giving input with stylus is gaining quite popularity. On-line recognition seems to be a simpler problem since more information is available. A few studies on converted or independent on-line and off-line data suggest superior recognition performance for on-line data [3]. The distinction between on-line and off-line recognition is not as rigid as it may seem.

The challenges posed by the online character recognition system are to increase the recognition accuracy and to reduce the recognition time. Recently some methods have been proposed that extract temporal information from static off-line data. This would allow for on-line treatment of off-line data. Also on-line systems have been designed using spatial representations of the on-line data, or combining on-line and off-line representations by supplementing each on-line data point with a pixel image of its local surroundings.

On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The obtained signal is converted into letter codes which are usable within computer and text-processing applications. The elements of an on-line handwriting recognition interface typically include a pen or stylus for the user to write with, a touch sensitive surface, which may be integrated with, or adjacent to, an output display and a software application which interprets the movements of the stylus across the writing surface, translating the resulting strokes into digital text.

1. *Direction Based Algorithm*

Many online character recognition methods use directional information to recognize a character [14], which assumes that same directional information is generated when written again. For example take only four directions that is right (r), down (d), left (l) and up (u) which helps in modeling strokes using regular expressions.

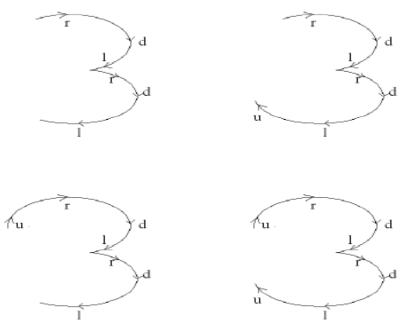

Fig. 2 Various Signatures of The First Stroke Of Character अ

The first stroke of अ has four different directional variations which can be modeled using regular expression $r.e(u|\lambda)rdlrdl(u|\lambda)$.

Fig. 2 shows the various signatures of first stroke of character अ .Ambiguities are resolved using spatial properties. It can be seen that success of the direction based approach is due to the less directional variations between the similar strokes. The number of variations depends upon the complexity in writing a stroke.

2. K-NN Classifier and DTW-Based Dissimilarity Measure

Classification is carried out by evaluating the dissimilarity measures [14] between the preprocessed input character and all the training samples and then applying the nearest neighbor rule. Fig. 3 shows the elastic matching of the characterक .

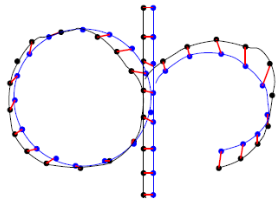

Fig.3 Elastic Matching for character क

Its major drawback is that it is computationally heavy, especially with large prototype sets and complicated similarity measures. For similarity measure it uses DTW measure between two strokes. DTW is an elastic matching technique that gives distance measure between character pairs. A stroke is a sequence of sample points from pen down to pen up events. The distance between strokes is calculated by considering all possible alignments between them, and finding the alignment for which the total distance is minimum using dynamic programming.

B. Off-line recognition

Off-line recognition operates on pictures generated by an optical scanner. The data is two-dimensional and space-ordered which means that overlapping characters cannot be separated easily. Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting. Off-line handwriting recognition is comparatively difficult, as different people have different handwriting styles. And, as of today, OCR engines are primarily focused on machine printed text and ICR for hand "printed" text.

Cursive handwriting utilizes the Hough transform and a neural network [4]. The Hough transform is a line detection technique which has the ability of tolerating deformation, disconnections and noise. Instead of searching for linear strokes in the image, the global directional information at each pixel of the image is computed. This information is stored into several feature maps. Assigning to each pixel a single orientation is avoided in order to preserve useful information. Each feature map is then processed by zones in order to estimate the local orientation of the strokes. Finally, image is recognized by means of a neural network classifier. These systems work for the recognition of segmented cursive characters, cursive words and the first letter of cursive words.

There are simple and fast algorithms for detection of italic, bold and all-capital words without doing actual character

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

recognition [3]. Researchers present a statistical study which reveals that the detection of such words may play a key role in automatic Information Retrieval from documents. Moreover, detection of italicized words can be used to improve the recognition accuracy of a text recognition system. Considerable numbers of document images have been tested and these algorithms give accurate results on all the tested images, and the algorithms are easy to implement. The four important methods in this category are Clustering, Feature Extraction, Pattern Matching and Artificial Neural Network.

1. Clustering

The goal of a clustering analysis is to divide a given set of data or objects into a cluster, which represents subsets or a group. The partition should have two properties. Homogeneity inside clusters: the data, which belongs to one cluster, should be as similar as possible. Heterogeneity between the clusters: the data, which belongs to different clusters, should be as different as possible [7].

The membership functions don't reflect the actual data distribution in the input and the output spaces. They may not be suitable for fuzzy pattern recognition. To build membership functions from the data available, a clustering technique may be used to partition the data, and then produce membership functions from the resultant clusters. Thus, the characters with similar features are in one cluster. Thus, in recognition process, the cluster is identified first and then the actual character.

1. K-means Algorithm

K-means is a simple unsupervised learning method which can be used for data grouping or classification when the number of the clusters is known. Thus, this method works for a fixed set of characters. Given a set of initial clusters, assign each point to one of them, and then each cluster centre is replaced by the mean point on the respective cluster. These two simple steps are repeated until convergence. A point is assigned to the cluster which is close in Euclidean distance to the point. Although K-means has the great advantage of being easy to implement, it has two big drawbacks. First, it can be really slow since in each step the distance between each point to each cluster has to be calculated, which can be really expensive in the presence of a large dataset. Second, this method is really sensitive to the provided initial clusters, however, in recent years, this problem has been addressed with some degree of success.

$$d(x,y) = \sum_{i=1}^{n} |y_i - x_i|$$

A variation of K-means is obtained and it is called **K-median**. Sometimes, this variation is less sensitive to outliers than traditional K-means due to the characteristics of the 1-norm.

The algorithm is as follows:

1. Select k objects as initial centres;
2. Assign each data object to the closest canter;
3. Recalculate the centres of each cluster;
4. Repeat steps 2 and 3 until centres do not change;

The main weakness points of K-means are the number of clusters may or may not be known prior and the randomly initialization of clusters seed effect the result of clustering. Fig. 4 shows two clusters 1 and 2 formed using k-means algorithm according to their centroids.

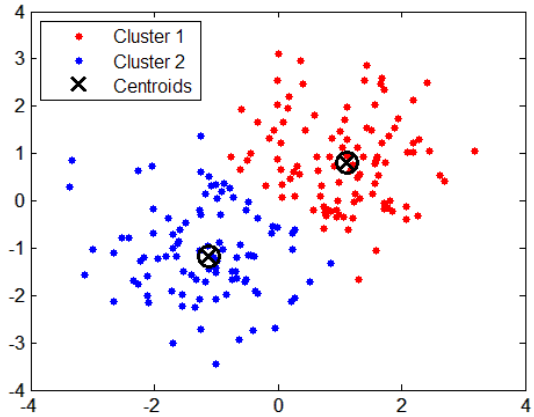

Fig. 4 K-means clustering

2. Hierarchial Algorithms

The hierarchical clustering of documents can be carried out either divisively or agglomerative. Divisive clustering breaks one complete cluster down into smaller pieces. In agglomerative clustering individual item similarities are used as a starting point and a gluing operation collects similar items, or groups, into larger groups.' Using these techniques, classes of similar objects are basically found by doing pairwise comparisons among all of the data elements. These clustering algorithms are serial in nature in that pairwise comparisons are made one at a time and the classification structure is created in a serial order.

The execution time for clustering the database based on a binary field using the agglomerative algorithms are more or less equal and the execution time increases as the size of the database increases [12]. However, divisive algorithm require lesser time than agglomerative algorithms when the size of the database increases.

1. Agglomerative Algorithm

For n samples, agglomerative algorithms [12] begin with n clusters and each cluster contains a single sample or a point. Then two clusters will merge so that the similarity between them is the closest until the number of clusters becomes 1 or as specified by the user.

1. Start with n clusters, and a single sample indicates one cluster.
2. Find the most similar clusters and then merge them into one cluster.
3. Repeat step 2 until the number of cluster becomes one or as specified by the user.

The distances between each pair of clusters are computed to choose two clusters that have more opportunity to merge.

2. Divisive Algorithms

Divisive algorithms [12] begin with just only one cluster that contains all sample data. Then, the single cluster splits into 2 or more clusters that have higher dissimilarity between

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

them until the number of clusters becomes number of samples or as specified by the user. The following algorithm is one kind of divisive algorithms using splinter party method.

1. Start with one cluster that contains all samples.
2. Calculate diameter of each cluster. Diameter is the maximal distance between samples in the cluster. Choose one cluster C having maximal diameter of all clusters to split.
3. Find the most dissimilar sample x from cluster C. Let x depart from the original cluster C to form a new independent cluster N (now cluster C does not include sample x). Assign all members of cluster C to MC.
4. Repeat step 6 until members of cluster C and N do not change.
5. Calculate similarities from each member of MC to cluster C and N, and let the member owning the highest similarities in MC move to its similar cluster C or N. Update members of C and N.
6. Repeat the step 2, 3, 4, 5 until the number of clusters becomes the number of samples or as specified by the user.

3. Self Organizing Map (SOM) Algorithm

Although computationally extensive, this class of techniques, which is based on network data structures and statistical algorithms, is generally flexible and powerful and is suited for parallelization. The self-organizing feature map (SOM) algorithm, developed by Kohonen [9][10], in particular, has been widely used in many different engineering and scientific applications such as image recognition, signal processing, and connectionist natural language processing. In addition, SOM is also widely used in visualization as a dimension (feature) reduction tool.

This network contains two layers of nodes - an input layer and a mapping (output) layer in the shape of a two-dimensional grid. The input layer acts as a distribution layer. The number of nodes in the input layer is equal to the number of features or attributes associated with the input. Each node of the mapping layer also has the same number of features as there are input nodes. Thus, the input layer and each node of the mapping layer can be represented as a vector which contains the number of features of the input. The network is fully connected in that every mapping node is connected to every input node. The mapping nodes are initialized with random numbers. Fig. 5 shows the Kohonen SOM topology. [10]

Each actual input is compared with each node on the mapping grid. The ``winning" mapping node is defined as that with the smallest Euclidean distance between the mapping node vector and the input vector. The input thus maps to a given mapping node. The value of the mapping node vector is then adjusted to reduce the Euclidean distance. In addition, all of the neighbouring nodes of the winning node are adjusted proportionally. In this way, the multi-dimensional (in terms of features) input nodes are mapped to a two-dimensional output grid. After all of the input is processed (usually after hundreds or thousands of repeated presentations), the result should be a spatial organization of the input data organized into clusters of similar (neighbouring) regions. Many engineering and scientific applications which involve numeric data (e.g., image recognition, signal processing) have successfully adopted the SOM approach to parallel clustering.

Training instances are often presented multiple times and network performance is achieved only after gradual modification of network connection/link weights. The robustness of the SOM algorithm and its appealing visualization effect has also made it a prime candidate in several large-scale information categorization and visualization projects [10].

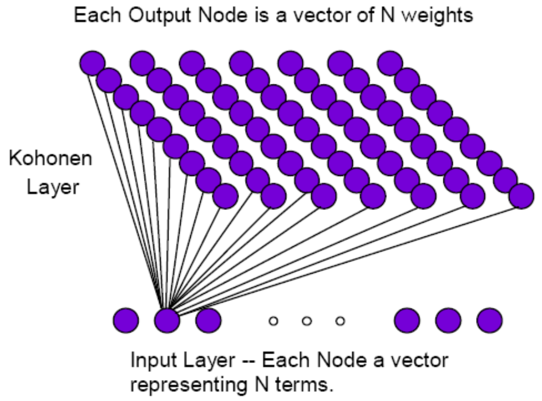

Fig. 5 Kohenen SOM Topology

4. Expectation Maximization (EM) Algorithm

It is used for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. Expectation maximization (EM) is a well-known algorithm used for clustering in the context of mixture models [9]. EM was proposed by Demster. This method estimates missing parameters of probabilistic models. Generally, this is an optimization approach, which had given some initial approximation of the cluster parameters, iteratively performs two steps: first, the expectation step computes the values expected for the cluster probabilities, and second, the maximization step computes the distribution parameters and their likelihood given the data. It iterates until the parameters being optimized reach a fix point or until the log-likelihood function, which measures the quality of clustering, reaches its maximum. The algorithm is similar to the K-means procedure in that a set of parameters are re-computed until a desired convergence value is achieved.

A mixture is a set of N probability distributions where each distribution represents a cluster. An individual instance is assigned a probability that it would have a certain set of attribute values given it was a member of a specific cluster. In the simplest case N=2, the probability distributes are assumed to be normal and data instances consist of a single real-valued attribute. Using the scenario, the job of the algorithm is to determine the value of five parameters, specifically:

1. The mean and standard deviation for cluster 1

2. The mean and standard deviation for cluster 2

3. The sampling probability P for cluster 1 (the probability for cluster 2 is 1-P). and the general procedure states as follow:

1. Guess initial values for the five parameters.

2. Use the probability density function for a normal distribution to compute the cluster probability for each instance. In the case of a single independent variable with mean µ and standard deviation σ, the formula is:

$$f(x)=\frac{1}{(\sqrt{2\pi\sigma})e^{\frac{-(x-\mu)^2}{2\sigma^2}}}$$

In the two-cluster case, we will have the two probability distribution formulas each having differing mean and standard deviation values.

3. Use the probability scores to re-estimate the five parameters.

4. Return to Step 2.

The algorithm terminates when a formula that measures cluster quality no longer shows significant increases. One measure of cluster quality is the likelihood that the data came from the dataset determined by the clustering. The likelihood computation is simply the multiplication of the sum of the probabilities for each of the instances. With two clusters A and B containing instances x1, x2, x3,…,xn where PA=PB=0.5 the computation is:

$$[.5P(x_1|A)+.5(x_1|B)][.5P(x_2|A)+.5(x_2|B)]..[.5P(x_n|A)+.5(x_n|B)]$$

After data pre-processing and exploratory data analysis have been completed, we can finally begin the modelling phase. This is the favourite part for many web usage miners, since it allows them to apply the range of their data mining skills and attack the problem at hand using an array of data mining methods, algorithms, and models. For modelling of user navigation patterns, we need to apply training dataset to specific algorithm.

C. Feature Extraction

The idea of the feature point extraction algorithm is to identify characters based on features that are somewhat similar to the features humans use to identify characters [1][8][16]. Programmers must manually determine the properties they feel are important. Some example properties might be Aspect Ratio, Percent of pixels above horizontal half point, Percent of pixels to right of vertical half point , Number of strokes , Average distance from image centre , Is reflected y axis , Is reflected x axis. Researchers have used many methods of feature extraction for handwritten characters [5]. Shadow code, fractal code, profiles, moment, template, structural (points, primitives), wavelet, directional feature etc., have been addressed in the literature as features. From the literature survey of the existing pieces of works on characters recognition, it was evident that not much effort is given on feature enhancement to remove the confusion between similar shaped characters for their recognition.

This approach gives the recognizer more control over the properties used in identification. Yet any system using this approach requires substantially more development time than a neural network because the properties are not learned automatically. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition systems.

1. Projection Method

The projection method [13] does the compression of the data through a projection. Black pixel counts are taken along parallel lines through the image area to generate marginal distributions. The direction of projection can be horizontal axis, vertical axis, diagonal axis or all of the above. Evermore, the character can be divided vertically and horizontally into four parts and do the same projection on each quarter. It will improve the recognition rate. Fig. 6 [13] shows horizontal and vertical projection of a character.

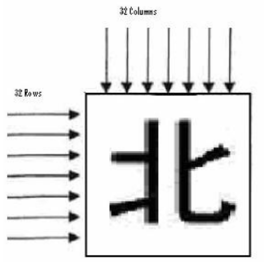

Fig. 6 Projection method

2. Border Transition Technique (BTT)

Border transition technique assumes that all the characters are oriented vertically. Each character is partitioned into four equal quadrants. The scanning and calculation of zero-to-one transition in both vertical and horizontal directions in each quadrant take place. Fig. 7 shows the partition and transition of a character 6 using BTT.

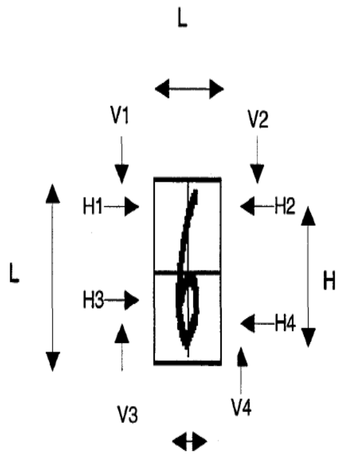

Fig.7 Border Transition Technique

1. Zoning

Zoning is a method involves the division of the character into smaller fragment of areas (zone) [13]. The black pixels in each zone are counted and accumulating or averaging the profiles in each zone extracts features. Fig. 8 shows the 16X16 and 8X8zoning.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

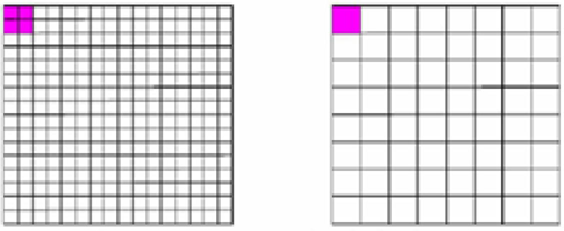

Fig. 8 16X16 to 8X8 zoning

2. Graph Matching Method

A graph matching method [4] uses structural feature of character. It is robust method to change of font or rotation. Three features are defined. First, an end point is connected only one pixel which has information of position. A branch point is connected more than three pixels. It has feature information which is connected the branch point. The information includes kind of features, position and direction. And a curve point is connected two pixels. However a straight line is also connected two pixels. In order to discriminate between a curve point and a straight line, direction information is used.

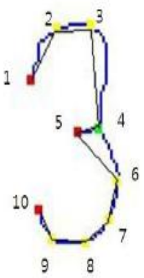

Fig. 9 Graph matching method

Fig. 9 shows character 3 on which Graph Matching Method is applied and it is described using end points, branch point and curve point.

D. Pattern Matching

In Pattern Matching, a character is identified by analysing its shape and comparing its features that distinguish each character. Various handwritten characters from forms or peripheral devices etc. are recognized with the help of various pre-processing and image enhancement techniques. These characters are further more specifically recognized by Pattern matching using Neural Network.

Pattern matching can be subdivided into two categories, depending on the type of input patterns: photometric pattern matching and geometric pattern matching. Photometric methods work directly on images which are considered as arrays of intensity values or real valued functions. Geometric methods work on geometric data such asnite point sets or polygons. This geometric data may be obtained directly from vector based object representations. It may also be obtained from images using feature extraction techniques. In this context, geometric pattern matching may be called feature based pattern matching. Fig. 10 shows different strokes in various directions. The character image is defined in terms of the combination of these strokes.

Fig.10 Synthesized Strokes in Varying Direction

The template matching has high speed, but is not very effective when there are font discrepancy, font slant, font defilement, stroke connection and stroke breaking due to the environment or the instrument itself. The neural network has relatively great space to enhance this recognition effect, which can accomplish higher recognizing ratio with more training. But the speed of entire system may become much slower due to its slow recognition rate. Therefore, a new solution that combines the advantages of both methods is used in some systems.

In feature extraction, normalizing the character image into standard size, directional planes is generated each recording the local stroke components in a specific direction or orientation [15]. To reduce the dimensionality of features, each directional plane is partitioned into a number, say, 8 by 8, of blocks, and the feature strengths of each block are averaged. To smooth the margins of blocks, the directional planes are better reduced by low pass filtering and down-sampling.

1. Directional Pattern Matching

The extraction of discriminative features is crucial to the recognition performance of pattern matching. Though sophisticated classification is now available for character recognition, pattern matching still survives for, e.g., coarse classification, and the evaluation of features with pattern matching makes sense for various classifiers. Desirable similarity measure should have the following properties:

- Continuity: when patterns are given small shape variations, the change of similarity measure should be small.
- Such variations can be parallel translation, stroke thickness changes, and angular (directional) changes.
- For character recognition, the similarity should decrease with the increasing angle between strokes. Particularly, two strokes that are perpendicular with each other should have a very small similarity.
- The similarity should be insensitive to the change of stroke width when other differences (direction, position, etc.) are small.

Representing two patterns in feature vectors x1 and x2 , the similarity is calculated as the correlation coefficient

$$Simi(\mathbf{x}_1, \mathbf{x}_2) = \frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{\| \mathbf{x}_1 \| \cdot \| \mathbf{x}_2 \|}$$

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

E. Artificial Neural Network

The main driving force behind neural network research is the desire to create a machine that works similar to the manner our own brain works. Neural networks have been used in a variety of different areas to solve a wide range of problems. Unlike human brains that can identify and memorize the characters like letters or digits; computers treat them as binary graphics. Therefore, algorithms are necessary to identify and recognize each character [13]. A neural network is a processing device, either an algorithm or an actual hardware, whose design was inspired by the design and functioning of animal brains and components thereof. The neural networks have the ability to learn from example, which makes them very flexible and powerful. These networks are also well suited for real-time systems because of their fast response and computational times, which are because of their parallel architecture.

A neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input values. Neural Network learning is also known as connectionist learning due to the connection between units. Fig.11 shows the mathematical representation of the ANN.

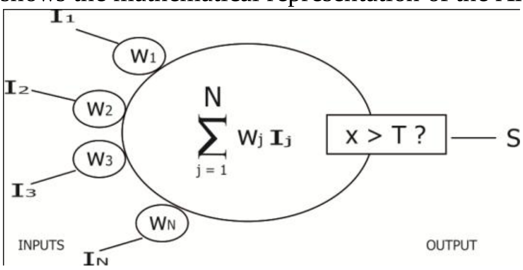

Fig.11 Mathematical Representation of ANN

Neural network recognizers learn from an initial image training set. The trained network then makes the character identifications. Each neural network uniquely learns the properties that differentiate training images. It then looks for similar properties in the target image to be identified. Neural networks are quick to setup; however, they can be inaccurate if they learn properties that are not important in the target data.

1. Backpopagation Algorithm

A Back propagation (BP) network consists of at least three layers of units, an input layer, at least one intermediate hidden layer, and an output layer. [13] When a Back propagation network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden-to-output weights. Fig.12 shows the working of BP algorithm. Back propagation learns by iteratively processing a data set of training values, comparing the network's prediction for each set with the actual known target values [6]. For each training dataset, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the "backwards" direction, i.e. from the output layer, through each hidden layer. Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

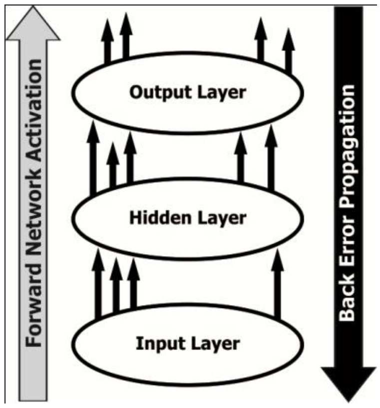

Fig. 12 Working of Back propagation Algorithm

Neural network uses Back propagation which is a technique and a supervised algorithm that learns by first computing the output using a feed forward network, then calculating the error signal and propagating the error backwards through the network [16]. The BP algorithm's most specific feature is the error that the neural network gets on its output. This error is equal with the difference between the real and required output values. When using BP, the authors look at the weight/change, which mostly minimizes the error in the output [2]. This algorithm is a method that depends on the gradient value of the moment. The learning starts when all of the training data was showed to the network at least once. The learning method, as for every network learning algorithm, consists of the modification of the weights. The BP algorithm consists of the following steps:

1. Define a training-sample for the network.
2. Compare the gotten output value with the required ones, and calculate the error for every output neuron.
3. Calculate the required output for every single neuron. The authors must also take into consideration the incremental factor, which shows us how much every neuron weight has to be changed, so that they will be perfect in values. This shows the local error.
4. Modify the weight in from of every neuron in the way to minimize the local error.
5. Give a level of blame to every neuron, this ay giving higher responsibility for those neurons with greater weights before them.
6. Repeat the method from step 3 for the neurons of the previous layer, using the "blame" as factor.

A popular and simple NN approach to the OCR problem is based on feed forward neural networks with BP learning. In the training step, the authors each training sample is represented by two components: possible input and the desired network's output given that input. Neural networks have high tolerance of noisy data and the ability to classify patterns on which they have not been trained. [6] They may be used when the authors have little knowledge of the relationships between attributes and classes. They are well suited for continuous-valued inputs and outputs, unlike most decision tree algorithms. They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text. Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process. Several techniques also have been developed for extraction of rules from trained neural network. These factors contribute toward the usefulness of neural networks for classification and prediction applications.

Neural network requires long training times and are therefore more suitable for applications where this is feasible [6]. They require a number of parameters that are best determined empirically, such as network topology or structure. Neural networks have been criticized for their poor interpretability.

2. RBFNN

As Fig. 13 shows, the RBFNN is composed of an input layer, a concealed layer and an output layer, in which the input layer and the output layer are respectively corresponding to the input vector space and classifications [17]. The concealed layer is made up of radial basis function nodes. Gauss kernel function is usually selected as the basis function.

$$\phi_l(x) = \exp\left(-\frac{\left\| x - c_l E \right\|}{2\sigma_l^2}\right)$$

Where x is the input vector, c_l is the kernel of the l th Gauss unit, E is an unit column vector of n dimension, σ_l is the radius.

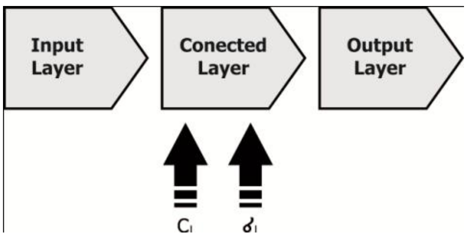

Fig. 13 RBFNN Structure

Output of the k th node of the output layer is

$$y_k = \sum_{l=1}^{h} w_{kl}\phi_l(x)$$

w_{kl}, where h is the number of the nodes of the concealed layer, is the linear weighted sum from the l th node of the concealed layer to the k th node of the output layer. The learning process of RBF arithmetic includes the learning process of the concealed layer and the output. The arithmetic of K-mean clustering is used in the learning process of the concealed layer, in order to get the kernel j c and the radius j. The method of gradient descent is used in the learning process of the output layer, in order to get the weight matrix W from the concealed layer to the output layer. RBFNN has random accuracy of functional approach ability, optimal characteristic of functional approach, and a good constringency.

3. Parallel BP Network

Among many neural networks arithmetic in character recognition, BP arithmetic is the most robust one. In this paper, a parallel BP neural network is used to recognize characters [3]. A simple BP neural network consists of three layers: input layer, hidden layer and output layer. Among them the input of the nodes of the input layer are the pixels of character images, the output of the nodes of the output layer are 35 characters (for letter aggregation are 26 letters), and the number of nodes in hidden layer is determined by experiments, usually around the square root of the sum of the number of input nodes and output nodes. The main principles of BP neural networks can be concluded as following.

1) Input the character image into the BP neural networks.

2) Calculate the Error Function by comparing the recognizing result of the neural networks and the known character result so as to adjust the value of the connective parameters between layers and make the output of the network more approximate to the known result.

3) Train the networks with a series of known images with the purpose of optimizing the parameters of the networks.

Experimental results show that a simple BP neural network is not efficient enough in character recognition with a recognition rate of less than 90%. The direct reason is the loss of spatial compounding information. In order to improve the performance of the arithmetic, the authors choose a parallel BP neural network in this paper to make the arithmetic more efficiently. [3] The parallel BP neural network consists of two simple BP neural networks: BP neural networks A and BP neural networks B. In the input layer of neural networks A, the character image is input in a row-first way so that the row spatial information of the character is preserved. And in the input layer of neural networks B, the character image is input in a column-first way so that the column spatial information is preserved. Therefore, the proposed parallel BP neural networks can preserve the spatial compounding information and can effectively improve the recognition rate. Fig. 14 shows the architecture of the parallel BP neural network.

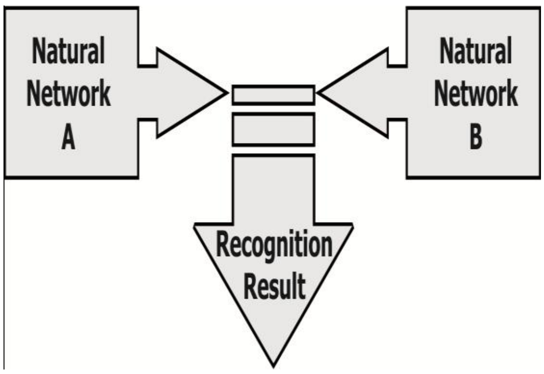

Fig. 14 The Parallel BPNN

Table 1 Overview of Various Character Recognition Methods

	Clustering	Feature extract-tion	Pattern matching	Neural Network
Operation	Divide data into groups	Find attributes input to classifier eg- aspect ratio	Analyze shape and compare features	Formulate rules for classifica-tion
Advantage	Grouping of data is easy	Features human use	High speed	Noisy data Learns
Scope	Confusing char	Not flexible	Need huge data	complexity, time
Applica-tion	Classifica-tion	Image analysis	Image database available	Incomplete database

III .CONCLUSION

The character recognition methods have developed remarkably in the last decade. A variety of techniques have emerged, influenced by developments in related fields such as image recognition and face recognition. In this paper, we have proposed an organization of these methods under two basic strategies. It is hoped that this comprehensive discussion will provide insight into the concepts involved, and perhaps provoke further advances in the area. The difficulty of performing accurate recognition is determined by the nature of the text to be read and by its quality. Generally, improper segmentation rates for unconstrained material increase progressively from machine print to handprint to cursive writing.

Current research employs models not only of characters, but also words and phrases, and even entire documents, and powerful tools such as HMM, neural nets, contextual methods are being brought to bear. While we have focused on the recognition problem it is clear that segmentation and classification have to be treated in an integrated manner to obtain high reliability in complex cases. Table 1 shows the overview of the various character recognition methods. The paper has concentrated on an appreciation of principles and methods. We have not attempted to compare the effectiveness of algorithms, or to discuss the crucial topic of evaluation. In truth, it would be very difficult to assess techniques separate from the systems for which they were developed. We believe that wise use of features and neural networks has led to improved accuracies. Features of each character are required based on which a character can be classified. Neural Network helps the system to recognize the character even if the exact pattern is not available in the database. The template matching has high speed, but is not very effective when there are font discrepancy, font slant, font defilement, stroke connection and stroke breaking due to the environment or the instrument itself. We can combine two or more techniques so as to improve the accuracy of the system. We have included a list of references sufficient to provide a more-detailed understanding of the approaches described. We apologize to researchers whose important contributions may have been overlooked.

ACKNOWLEDGEMENT

The authors wish to thank the anonymous reviewers for their useful suggestions that helped in improving the quality of this paper. This work was supported in part by MAEER's MIT, Pune and VIIT, Pune.

REFERENCES

[1] Dr. P. S. Deshpande, Mrs. Latesh Malik, Mrs. Sandhya Arora, "Handwritten Devanagari Character Recognition Using Connected Segments and Minimum Edit Distance" IEEE 2007.

[2] Rókus Arnold, Póth Miklós, "Character Recognition Using Neural Networks", CINTI 2010, 978-1-4244-9280-0/10/$26.00 ©2010 IEEE, 311-314.

[3] Feng Yanga, Fan Yangb, "Character Recognition Using Parallel BP Neural Network", ICALIP2008, pp 1595-1599, 978-1-4244-1724-7/08/$25.00©2008IEEE.

[4] Jieun Kim, Ho-sub Yoon, "Graph Matching Method for Character Recognition in Natural Scene Images", INES 2011, pp 347-350, 978-1-4244-8956-5/11/$26.00 ©2011 IEEE.

[5] T. Wakabayashi, U. Pal, F. Kimura and Y. Miyake, "F-ratio Based Weighted Feature Extraction for Similar Shape Character Recognition", ICDAR.2009, pp 196-200, 978-0-7695-3725-2/09 $25.00 © 2009 IEEE.

[6] E.Kavallieratos, N.Antoniades, N.Fakotakis and G.Kokkinakis, "Extraction and recognition of handwritten alphanumeric characters from application forms".

[7] Rumiana Krasteva, "Bulgarian Hand-Printed Character Recognition Using Fuzzy C-Means Clustering", Problems of engineering and robotics", pp 112-117.

[8] Mohammed Abu Ayshi, M.Jay Kimmel, Diane C. Simmons, "Character recognition system using spatial and structural features", US 7,010,166B2.

[9] Norwati Mustapha, Manijeh Jalali, Mehrdad Jalali, "Expectation Maximization Clustering Algorithm for User Modeling in Web Usage Mining Systems", European Journal of Scientific Research ISSN 1450-216X Vol.32 No.4 (2009).

[10] Dmitri G. Roussinov, Hsinchun Chen, "A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation".

[11] Shashank Araokar, "Visual Character Recognition using Artificial Neural Networks".

[12] Dr.N.Rajalingam, K.Ranjini, "Hierarchical Clustering Algorithm - A Comparative Study", International Journal of Computer Applications, 2011.

[13] Attaullah Khawaja, Shen Tingzhi, Noor Mohammad Memon, AltafRajpa, "Recognition of printed Chinese characters by using Neural Network", 1-4244-0794-X/06/$20.00 ©2006 IEEE, pp 169-172.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 5, May 2012

[14] K.Gupta, S.V.Rao, and P.Viswanath, "Speeding up Online Character Recognition", Proceedings of Image and Vision Computing New Zealand 2007.

[15] Hiromichi Fujisawa and Cheng-Lin Liu, "Directional Pattern Matching for Character Recognition Revisited", ICDAR 2003, 0-7695-1960-1/03 $17.00 © 2003 IEEE.

[16] Yuk Yirtg Chung, M'an To Wong, "Handwritten Character Recognition By Fourier Descriptors And Neural Network", 1997 IEEE TENCON, pp 391-394.

[17] Enyong Hu, Hui Wang, Jianhua Wang, Song Lu4, Lei Tian, "Study on pattern recognition model based on principal component analysis and radius basis function neural network", 978-1-4244-8728-8/11/$26.00 ©2011 IEEE, pp 388-390 .