An Approach for Character Recognition Using Pattern Matching with ANN

Anjali Chandavale, Suruchi Dedgaonkar, Dr. Ashok Sapkal

Abstract— Character recognition is the process to classify the input character according to the predefined character class. Recent computer applications should read the text, which may be in the form of scanned handwritten document or typed text in various fonts or a combination of both. The character recognition system must be faster and reliable. Thus, an algorithm is selected inspired from pattern matching and Artificial Neural Network (ANN). We have also implemented feature extraction and graph matching algorithms for analysing performance of the Pattern Matching and ANN method.

The experimental results show accuracy of the pattern matching and ANN algorithm is 93% for typed characters and 70% for the handwritten characters, provided the algorithm is trained with character set of at least 5 different fonts. The accuracy improves as ANN is trained with more patterns. Also the database size and response time of the Pattern Matching and ANN algorithm is lesser than the other two algorithms.

Index Terms— character recognition, feature extraction, pattern matching, graph matching, ANN, Classifier, handwritten character

———————————— ◆ ————————————

1 INTRODUCTION

THE same characters differ in sizes, shapes and styles from person to person and even from time to time with the same person. Like any image, visual characters are subject to spoilage due to noise near the edges. Also, there are no hard-and-fast rules that define the appearance of a visual character. Thus, classical methods in pattern recognition are not perfect for the recognition of visual characters [11].

The character recognition system consists of feature extractor and classifier with the stored patterns in the database as shown in Fig.1. The classifier assigns classes to the character. The character properties i.e. features serve the purpose of recognition.

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. Thus, there is need of the faster and reliable character recognition system.

2 LITERATURE SURVEY

There exist several different techniques for recognizing characters. Basically these methods are online or offline [1].

On-line recognition seems to be a simpler problem since more information is available [3],[14],[19]. Off-line recognition operates on pictures generated by an optical scanner. The offline methods are Clustering, Feature Extraction, Pattern Matching and Artificial Neural Network. [6] [19]

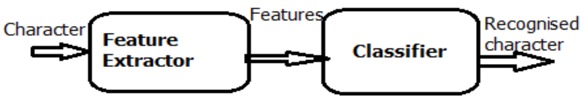


Fig.1 Main Character Recognition Steps

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, which are homogeneity inside clusters and heterogeneity between the clusters. [7],[9],[10],[12]

Feature extraction classifies the characters based on properties that are somewhat similar to the features humans use to identify characters [1],[8],[16]. Researchers have used many methods of feature extraction for characters [5]. This approach gives the recognizer more control over the properties used in identification.

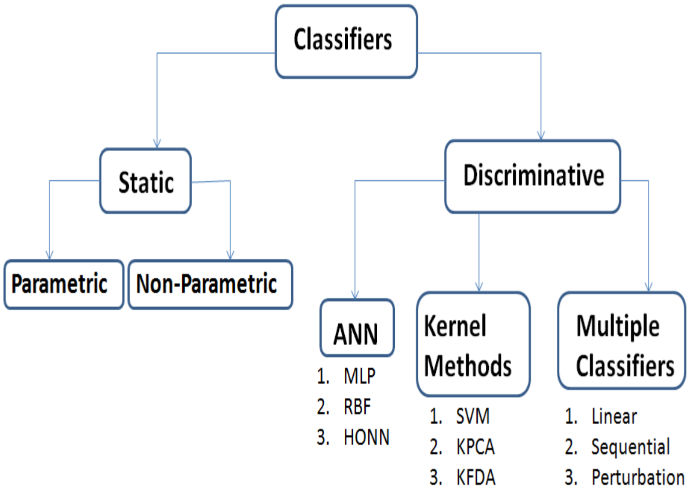


Fig. 2 Types of Classifiers

In Pattern Matching [15], a character is identified by analysing its shape and comparing its features that distinguish each character. Pattern matching methods are faster.

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category. Fig.2 shows that the classifiers are categorized as statistical methods, artificial neural networks, support vector machines and multiple classifier combination. Statistical Methods are based on Baye's rule. Discriminative methods are based on minimum error training.

The main driving force behind ANN research is the desire to create a machine that works similar to the manner our own brain works [2],[13],[16]. The neural networks have the ability to learn from examples, which makes them very flexible and powerful. ANN is also well suited for real-time systems because of their fast response and computational times which are because of their parallel architecture. Therefore this paper selects an approach of combining pattern matching and ANN. Thus the selected method is faster as well as accurate.

3 IMPLEMENTATION

The paper [19] has done the survey of various methods used for character recognition. We concluded saying that wise use of features and neural networks can lead 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. We can combine two or more techniques so as to improve the accuracy of the system. To verify the conclusion mentioned in [19], this paper has selected method inspired from pattern matching and ANN. The paper has also implemented feature extraction and graph matching so as to analyse the performance of the Pattern Matching and ANN method with feature extraction and graph matching algorithm.

3.1 Feature Extraction Method

In feature extraction, programmers must manually determine the properties they feel important. Some examples of properties are height, number of holes, maximum number of white-black transitions, nature of vertical stroke, aspect ratio, standard deviation, 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. [3]

After the input character is binarized and preprocessed, the features of the input character are found. (Fig.3) In the feature extraction algorithm, the features extracted are mean value, 8 values for horizontal zone mean and 8 values for vertical zone mean.

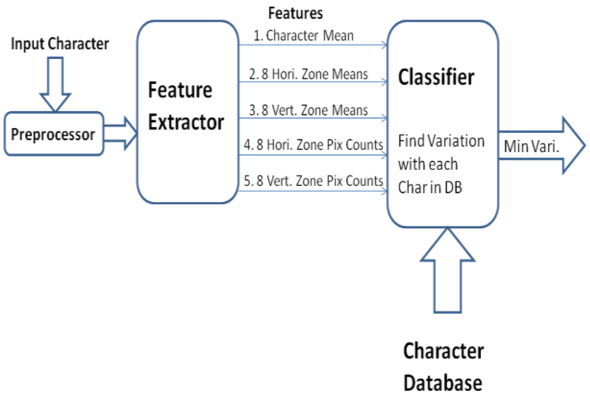


Fig.3 Functional diagram of Feature Extraction Algorithm

Recognition is done using standard deviation. In statistics and probability theory, standard deviation shows how much variation or "dispersion" exists from the average (mean, or expected value). A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data points are spread out over a large range of values. [18]

The parameters of one character class are estimated from the samples of its own character types only. Thus, the classifier used in this algorithm is parametric statistical classifier.

For recognition, we have used the standard deviation along x-axis and along y-axis as to compare primary features that represent the character properties. This value helps us to know the visual appearance of the character. Then, the algorithm to find squared standard deviation is as follows.

Step 1: Calculate the mean p1 of input character.

Step 2: Calculate the square of distance of mean of each data value p2 from the mean of input character p1. This is called Euclidean distance or variance (squared deviation) from mean and is given by

$$totalDist = d^2(p1,p2) = (y2 - y1)^2 - (x2 - x1)^2$$

Step 3: Similarly, update variance with all 8 horizontal zone means and 8 vertical zone means of input character

Step 4: Repeat the above steps for all character data in database.

Step 5: The character with the minimum variance with input character is the recognized character.

The limitation of this method is that it does not allow flexibility i.e. it does not give satisfactory results with multiple fonts at a time. The number of features used and the number of comparisons during recognition increases memory and processing time.

3.2 Graph Matching Method

A graph matching method [4] uses structural features as shown in Fig.4, of character. It is recognition method which considers relation of position and structural features. (Fig.5)

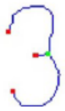


Fig.4 End points and branch points in a character

The features extracted are mean value, end points and branch points. We first find end points and branch points as shown in Fig.4. Then, find position information of end points and branch points.

1) End point and Branch point: We trace character lines using neighbour method. If pixel x satisfies condition of an end point or a branch point, pixel x is end point or branch point.

```
For all the four (n) sides of the current pixel,
    If (neighbour (current_pixel) = 0)
            count++;
```

Finally, if count value is 1, it is an end point and if count value is 3 then it is a branch point.

2) More information is required even though we have end point and branch point information, as there are number of characters having similar number of end points and branch points. Thus we find the distance between the end points and the branch points.

For recognition, we have to compare primary features that represent the character properties. The primary features are number of branch points and end points. If the primary features match, then only the method compares the secondary features viz. distance between the branch points and end points. The parameters of one character class are estimated from the samples of its own character types only. Thus, the classifier used in this algorithm is parametric statistical classifier. The steps for recognition are:

Step 1: Calculate the mean p1 of input character.

Step 2: Calculate the square of distance of mean of each data value p2 from the mean of input character p1. This is called Euclidean distance or variance (squared deviation) from mean and is given by

$$totalDist = d^2(p1,p2) = (y2 - y1)^2 - \ (x2 - x1)^2$$

Step 3: Similarly, update variance with variance of all end points and variance of all branch points of input character with each character in database.

Step 4: Repeat the above steps for all character data in database.

Step 5: The character with the minimum variance with input character is the recognized character.

This method also does not give satisfactory results for multiple fonts. But, as the number of features is less than the feature extraction method, the number of comparisons required is less.

Graph matching is a robust method. It works for multiple fonts and rotated characters. But if we try to input more than three different font styles, the accuracy of the method decreases.

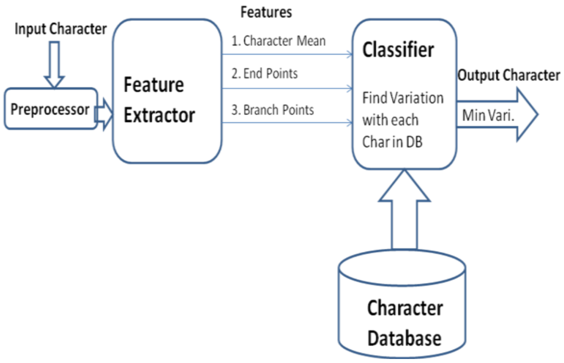


Fig. 5 Functional diagram of Graph Matching Algorithm

3.3 CR based on Pattern Matching and ANN

In both the above algorithms, the recognition is done by comparing input character with every character pattern in database. Thus, they are slow. Therefore, we have implemented a method based on Pattern Matching & ANN [11]. The pattern matching [15] [17] has high speed and the ANN has relatively great space to enhance this recognition effect, which can accomplish higher recognition ratio with more training. This method gives success rate as 93% which is improved as compared to feature extraction and graph matching.

The Neural Network is trained by entering different patterns of the character. These patterns are stored in the database. When any character is input for recognition, the recognition ratio is calculated which is ratio between input character with the saved character. The character with maximum recognition ratio is recognized and is output.

After the input character is binarized, we find the weight matrix of the given character. (Fig. 6) The system can be trained with multiple patterns for a single character. The Neural Network analyses multiple patterns for a character. The weight matrix is created for all the characters, which represents all input patterns of that character collectively. It is updated every time a letter is taught to the ANN, by incrementing similar segment value and decrementing distinct segment value. The weight matrix becomes more perfect as we teach more patterns.

When any character is input for recognition, the recognition ratio is calculated which is ratio between input character with the character stored in database. The connecting weights are adjusted to minimize the squared error on training samples in supervised learning. Thus, the classifier used in this algorithm is ANN discriminative classifier. The classifier outputs the character with maximum recognition ratio.

1. Several variant patterns of the same character are taught to the network under the same label. Hence the network

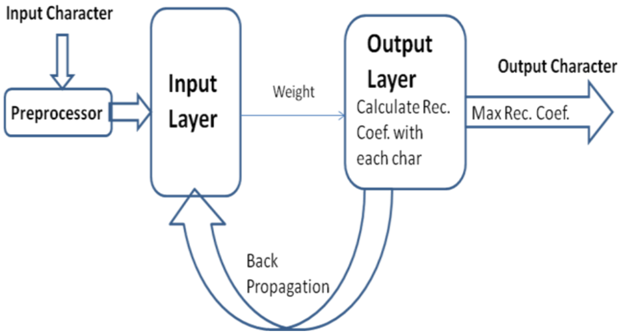


Fig.6 Functional diagram of the Pattern Matching and Algorithm

learns possible variations of a single pattern and becomes adaptive in nature. Each element of matrix M is defined to be 1 or -1 depending on pixel value. If pixel val=1, M element=1. If pixel val=0, M element= -1.

2. The input matrix M is fed as input to the neural network. The NN learns in a supervised manner by adjusting its weights.

3. In the method of learning, each candidate character taught to the network possesses a corresponding weight matrix. As learning of the character progresses, the weight matrix is updated
4. The matrix is initialized to zero. Whenever a character is to be taught to the network, an input pattern representing that character is submitted to the network. The weight matrix is updated for each sample of character k as

$$W_k(i,j) = W_k(i,j) + M(i,j)$$

5. The Candidate Score is a product of corresponding elements of the weight matrix Wk of the kth learnt pattern and an input pattern I as its candidate.

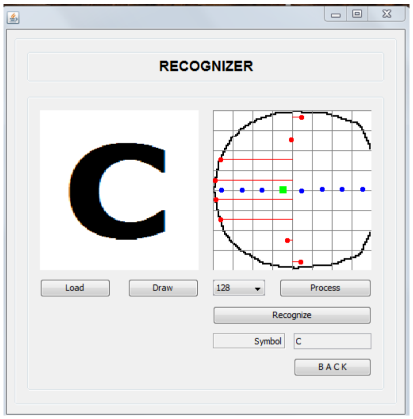
$$\psi(k) = \sum_{i=1}^{x} \sum_{j=1}^{y} W_k(i,j) * I(i,j)$$

6. µ is the ideal weight model score. The Recognition Quotient, Q is a measure of how well the recognition system identifies an input pattern as a matching candidate for one of its many learnt patterns. The greater the value of Q, the more confidence does the system bestow on the input pattern as being similar to a pattern already known to it.

$$Q(k) = \frac{\psi(k)}{\mu(k)}$$

It can be observed that by regular training, the system develops its ability to identify a matching pattern and reject nonmatching patterns. Thus, regular supervised training enhances the performance of the system.

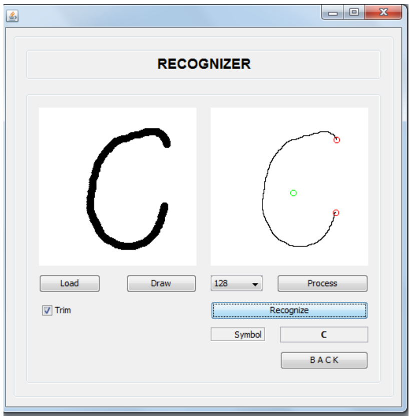
4 RESULTS



128

Symbol C

Fig.7 Character Recognition by Feature Extraction

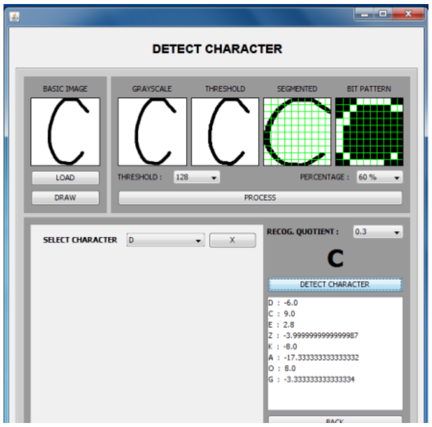


128

☑ Trim

Symbol C

Fig.8 Character Recognition by Graph Matching



THRESHOLD : 128

PERCENTAGE : 60 %

SELECT CHARACTER D

RECOG. QUOTIENT : 0.3

C

```
D : -6.0
C : 9.0
E : 2.8
Z : -3.9999999999999987
K : -8.0
A : -17.333333333333332
O : 8.0
G : -3.333333333333334
```

Fig.9 Character Recognition by Pattern Matching and ANN

The fig. 7, fig.8 and fig.9 show the working of feature extraction, graph matching and the suggested method respectively. The algorithm is tested for typed and handwritten lower and upper case characters each with different set of patterns.

We have trained all the three algorithms with 26 upper case alphabets and 0 to 9 digits of 5 different fonts viz. Times, Tahoma, Lucida, Copper Black and Arial Black. We have observed the results for the trained characters (known font) , untrained characters (unknown font) and handwritten characters.

It is observed that the feature extraction gives wrong results if the value of feature i.e. standard deviation is similar for the two characters like B and S. Also, graph matching gives wrong results for the characters having same end points, branch points and the distance between them. The graphs of 2 and R have same points.

The suggested system gives wrong results for the similar patterns like 1 and I, where recognition quotient value is almost similar. But after training the system with multiple patterns, it gives the correct result.

It is observed that the feature extraction gives wrong results if the value of feature i.e. standard deviation is similar for the two characters like B and S. Also, graph matching gives wrong results for the characters having same end points, branch points and the distance between them. The graphs of 2 and R have same points.

Pattern matching gives wrong results for the similar patterns like 1 and I, where recognition quotient value is almost similar. But after training the system with multiple patterns, it gives the correct result.

5 PERFORMANCE ANALYSIS

Accuracy of the algorithms is calculated by the following formula where T is total number of characters; TC is number of correctly recognized characters.

$$\text{Accuracy}= TC *100/ T$$

The accuracy of Pattern Matching and ANN method is 93% after training multiple patterns, whereas for feature extraction it is 85% and for graph matching it is 83%. Feature extraction and graph matching are useful for fixed font like typewritten characters. But, the suggested method can recognize handwritten characters due to its ability to learn and recognize new set of patterns. The Fig. 10 shows the analysis of accuracy of three algorithms for known font, unknown font and handwritten character.

It is also observed that the accuracy of the suggested system increases with the number of trained patterns. (Fig.11)

The accuracy of the suggested algorithm increases with the increased dimension of the weight matrix (Fig.12). But, it leads to increasing database size drastically. (Fig.13)

The size of the database file is more for feature extraction algorithm, as the number of features stored is more in number. Also, the database size of graph matching algorithm varies according to the number of end points and number of branch points found for that font. The database size of the Suggested algorithm is less, as a single weight matrix is used to store all the patterns of single character. (Fig.16)

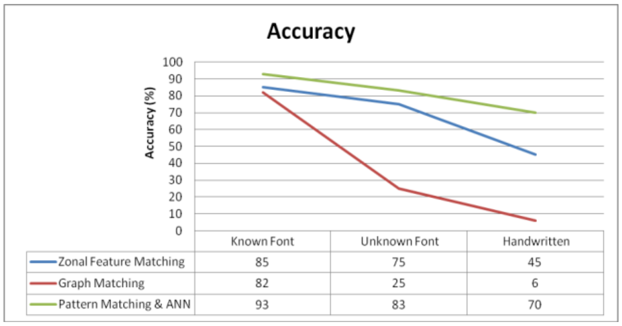


Fig.10 Analysis of Accuracy for Font Type by the Three algorithms

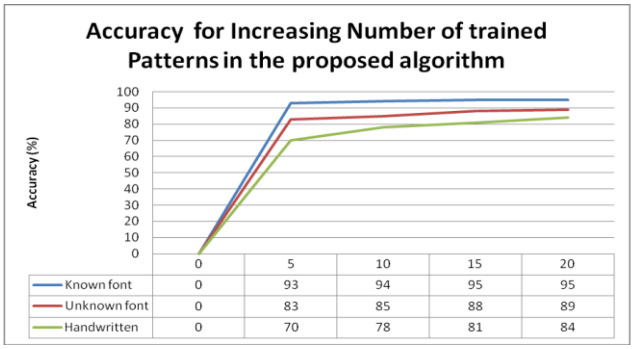


Fig.11 Analysis of Accuracy of the Suggested algorithm for the number of patterns trained

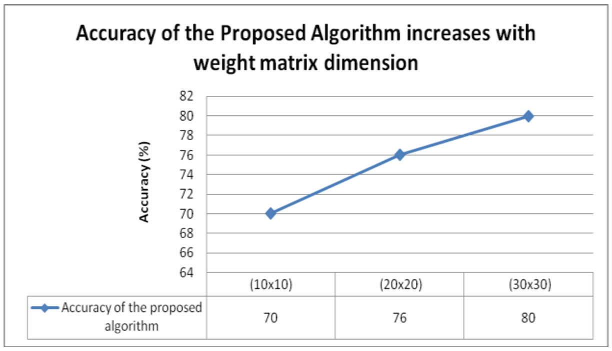


Fig.12 Analysis of Accuracy of the Suggested algorithm for the increasing weight matrix dimension

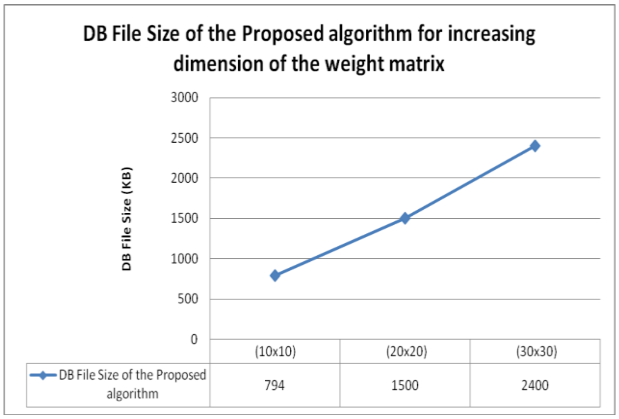


Fig.13 Analysis of Database Filesize of the Suggested algorithm for the increasing weight matrix dimension

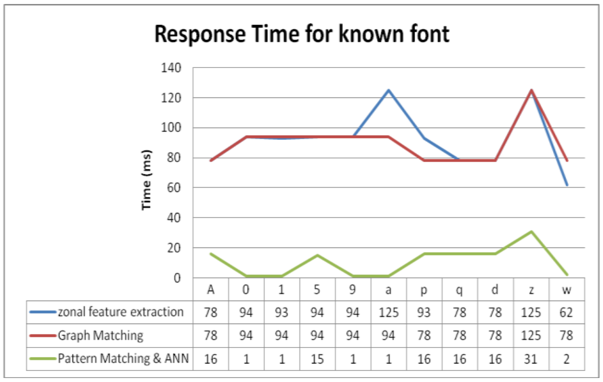


Fig.14 Analysis of Response Time for known fonts by three algorithms

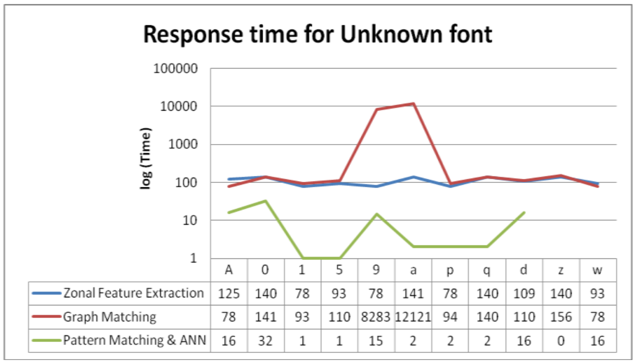


Fig.15 Analysis of Response Time for unknown fonts by three algorithms

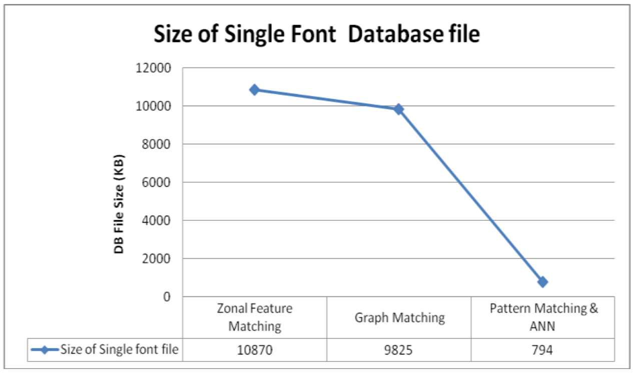


Fig.16 Analysis of Database File Size for single font by three algorithms

Fig.14 and Fig.15 show that the response time of the pattern matching and ANN algorithm is minimum for known as well as unknown fonts. Table 1 provides summary of the performance analysis.

TABLE 1
PERFORMANCE ANALYSIS OF THREE TECHNIQUES

	Feature Extraction	**Graph Matching**	**The Pattern Matching and ANN**
Size of Database Proportion al to	(no of fonts) x (no of chars)	(no of fonts) x (no of chars)	no of chars
Accuracy	Known: 85% Unknown: 75% Handwritten:45 %	Known: 75% Unknown: 25% Handwritten:6 %	Known: 93% Unknown: 83% Handwritten:70 %
Response Time	Consistent, more than pattern matching	Increases for unknown font characters	Minimum in all conditions
Advantage s	Simple, features are strong & comparable, Considers every pixel of character	Easy and robust, No need to access every pixel	Learns, recognizes unknown patterns, Single matrix for all patterns
Limitation	Size of database	Accuracy decreases with number of fonts	Learning time

6 Conclusion

We have implemented and compared the effectiveness of three algorithms which are Feature Extraction, Graph Matching and Pattern Matching with ANN. Feature extraction algorithm uses zonal features of character i.e. character mean, 8 horizontal zonal mean values, 8 vertical zonal mean values, number of pixels in 8 horizontal zones and number of pixels in 8 vertical zones. The graph matching algorithm uses less number of positional features i.e. character mean, branch points and end points. The results show that the graph matching and the feature extraction are suitable for the typewritten fonts as the fonts are fixed and less in number.

The CR based on pattern matching and ANN is faster and accurate. It is a universal approach as it recognizes the typed characters with 93% accuracy and handwritten characters with 70% accuracy. The result proves that the Pattern Matching and ANN is more accurate than the graph matching and feature extraction. But training ANN is complex and time consuming. It can detect typed as well as handwritten characters due to the ability of learning new patterns. The graph matching and the feature extraction are suitable for the typewritten fonts as the fonts are fixed and compact.

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.

ACKNOWLEDGEMENTS

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 Devnagari 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] A.A.Chandavale, Ashok M. Sapkal, Dr. R.M. Jalnekar, "Algorithm to break Visual CAPTCHA", ICETET-09

[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

[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

[18] Mamatha H.R.,Sucharitha S., Srikana Murthy K," Multi-font and Multi-size Kannada Character Recognition based on the Curvelets and Standard Deviation", International Journal of Computer Applications (0975 – 8887) Volume 35– No.11, December 2011

[19] Suruchi G.Dedgaonkar, Anjali A. Chandavale, Ashok M. Sapkal, "Survey of Methods for Character Recognition", ISSN:2277-3754, International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012