Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 79 (2016) 337 – 343

Procedia Computer Science

7th International Conference on Communication, Computing and Virtualization 2016

A comparative analysis of particle swarm optimization and support vector machines for devnagri character recognition: an android application

Prashant M. Kakde [a], Dr. S.M.Gulhane [b]
[a] Research Scholar, S.G.B.A.U Amravati, 444603,India,pmkakde@yahoo.com
[b] Prof.& Head of Department Electronics& Telecommunication Engg.
Jawaharlal Darda I.E.T Yavatmal,445001,India,Smgulhane67@rediffmail.com

Abstract

Devanagari script is widely used in the Indian subcontinent in several major languages such as Hindi, Sanskrit, Marathi and Nepali. Recognition of unconstrained (Handwritten) Devanagari character is more complex due to shape of constituent strokes. Hence character recognition (CR) has been an active area of research till now and it continues to be a challenging research topic due to its diverse applicable environment. As the size of the vocabulary increases, the complexity of algorithms also increases linearly due to the need for a larger search space. Devnagari script recognition systems 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. In this paper techniques like particle swarm optimization and support vector machines are implemented and compared. An android phone is used for taking input character and MATLAB software for showing the recognized Devnagari character. For the connection between android device and MATLAB we are using PHP language. The particle swarm optimization technique provides accuracy up to 90%.

Keywords: Handwritten Character Recognition (HCR); Feature detection; Support Vector Machine (SVM); Particle Swarm Optimization (PSO); Android device; MATLAB.

1. Introduction

Handwriting recognition (HWR) is an 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. Handwritten Character Recognition (HCR) is the process of classifying written characters into appropriate classes based on the features extracted from each character. Handwritten character recognition can be performed either online or offline. The character set of Indian languages is large and consists of more complex characters when compared to the Latin script. In past few years, computerization has taken over large number of manual operations. In the early stage of OCR 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. Handwritten character recognition, irrespective of the script, finds

Peer-review under responsibility of the Organizing Committee of ICCCV 2016
doi:10.1016/j.procs.2016.03.044

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.

New learning methods, using support vector machines (SVMs), particle swarm optimization (PSO) are now actively studied and applied in pattern recognition problems. Learning methods have beneficiated 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.

1.1Characteristics of Devnagri Script

Devnagari script is different from Roman script in many ways. Devnagari script has two-dimensional compositions of symbols i.e. core characters in the middle strip and optional modifiers above 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 scripts are 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 different shapes or patterns, which makes Devnagari OCR more difficult to develop. Figure 1 shows some of the characters of devnagri script.

Fig.1. Printed samples of Devnagari characters e.g. Vowels

2. 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. Moment based features are extracted from scaled, thinned one pixel wide skeleton of character image.

2.1 Chain Code Histogram of Character Contour

Given a scaled binary image, first find the contour points of the character image. We consider a 3 × 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.

	X	
X	**P**	**X**
	X	

Fig 2 Contour point detection

The contour following procedure generates a contour representation called “chain coding” proposed by Freeman. Each pixel of the contour is assigned a different code that indicates the direction of the next pixel that belongs to the contour in some given direction. In this methodology of using a chain coding of connecting neighboring contour pixels, the points and the outline coding are captured. Contour following procedure may proceed in clockwise or in counter clockwise direction. Here, we have chosen to proceed in a clockwise direction. 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 are multiple connectivity’s 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 × 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 × 5 × 8 = 200 chain code features for recognition.

2.2 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:-

$$\mu_{pq} = \sum \sum (x-x)^p (y-y)^q \ f(x,y) \ \ldots\ldots (1)$$

Where x, y are the co-ordinates of the region's center of gravity (centroid). These can be obtained using the following equations:

$$\bar{x} = \frac{m_{10}}{m_{00}} \ and \ \bar{y} = \frac{m_{01}}{m_{00}} \ldots\ldots (2)$$

The central moments of up to order 3 can be obtained from the above equation by choosing p, q = 0, 1, 2, 3 such that $p + q$ 3. The normalized central moments denoted by pq, are denoted by $pq = \mu pq / \mu 00\ y$ where $y = (p + q)/2 + 1$ for $p + q$ = 2, 3.... Rotation invariance can be achieved if the coordinate system is chosen such that μ11 = 0. Seven rotation, translation, and scale invariant moment characteristics can be derived from the second and third moments.

$$\emptyset_1 = n_{20} + n_{02} \ldots\ldots (3)$$

$$\emptyset_2 = (n_{20} - n_{02})^2 + 4n_{11}{}^2 \ldots\ldots (4)$$

$$\emptyset_3 = n_{30} + 3n_{12}{}^2 + (3n_{21} - n_{03})^2 \ldots\ldots (5)$$

$$\emptyset_4 = (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2 \ldots\ldots (6)$$

$$\emptyset_5 = (n_{30} - 3n_{12})(n_{30} + n_{12}) + [(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2 \ldots\ldots (7)$$

$$\emptyset_6 = (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \, 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \ldots\ldots (8)$$

The values of these seven moments for a given basic symbol image represent the basic symbol and are used to create a feature vector consisting of seven values. Image is segmented into nine equal sub-images and in each sub-images moment features are calculated so total 63 features are formed.

3. 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 hyper-plane with maximum margin between the two classes in the feature space[5]. This results in a nonlinear boundary in the input space. The optimal separating hyper-plane 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:

$$K(x, y) = x.y \quad (9)$$

2. Radial Basis Function (Gaussian) kernel:

$$K(x, y) = \exp(-||x - y||^2 / 2\sigma^2) \quad (10)$$

3. Polynomial Kernel:

$$K(x, y) = (x.y + 1)^d \tag{11}$$

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. First the classifier is trained with the 'n' number of input image sequences of the different character images. The features of these images are extracted by the algorithm and then these features are stored in the trained classifier. For testing the image first features of the testing image are extracted and these features are converted into feature vector and this feature vector is compared with the trained classifier of the SVM and gives the output result image.

.

4. Particle Swarm Optimization

Kennedy and Eberhart in 1995 designed an algorithm known as the Particle Swarm Optimization. PSO is a population-based searching method. PSO is a met heuristic technique, as it makes a few or no assumptions about the problem but will search very large spaces of the solution. Particle Swarm Optimization algorithm imitates the social behavior of fish schools or bird flocks. 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 3 [11] [12]. The parameter selection can have a large impact on optimization performance. They can also be tuned by using another overlaying optimizer, known as meta-optimization. It contends that the PSO algorithm and its parameters must be chosen so as to properly balance between exploration and exploitation to avoid premature convergence to a local optimum yet still ensure a good rate of convergence to the optimum. This belief is the precursor of many PSO variants.

$$V_{id}(t+1) = W * V_{id}(t) + C_1R_1(P_{id}(t) - X_{id}(t) - C_1R_1 P_{id}(t) - X_{id}(t) \tag{12}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{13}$$

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.

R1, R2 are separately generated random numbers in the range of zero and one.

W: is the inertia weight.

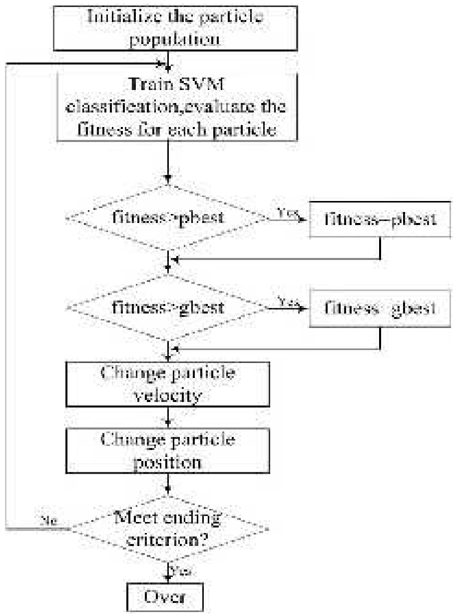

Fig 3. Flow chart of particle swarm optimization technique

5. Proposed System

In proposed system the online Devanagari Character Reorganization system is developed. In this system an android device is used for taking the input and Matlab software to show the recognized character as the output in the system on computer screen. For connecting android device to system PHP sever side scripting language is used. In an android device database all the features of the gestures of the Devnagari characters are stored. To find out the features of the gesture, an android provides the Gesture Recognition API. With the help of Gestures Recognition API features of the Devanagari characters are stored in the SQLite database of the android device. Matlab software is used to implement particle swarm optimization algorithm for the recognition of the characters. An Android database in which all features of the Devanagari characters are stored is the used for finding the best match according to the PSO algorithm. Android database in which all features of the Devanagari characters are stored is the used for training support vector machine. The neural network is used for the feature extraction and character recognition. SVM allow to 'n' number of possible features for the same image which helps to improve the performance of the recognition technique.

First the input Devanagari characters are given as the input from the android device. Feature extraction is done by the help of API provided by the android device. These features are then sent to the Matlab software from the PHP server. Image processing techniques are performed on the features in Matlab for the classification of the characters. Now PSO algorithm is used for the classification of characters from the input features. The classification result shows the best match characters from the database as an output on the Matlab screen.

6. Experimental Results

Table1 shows the comparative analysis of PSO and SVM techniques based on output accuracy of the character recognition with the time required to perform the operation.

Table 1

No. of Words	Accuracy		Time	
	SVM	PSO	SVM	PSO
10	85%	90%	50ms	55ms
20	83.23%	88.65%	90ms	93ms
30	78.45%	83.43%	140ms	135ms
40	76.54%	87.59%	179ms	183ms
50	74.21%	82.21%	200ms	210ms

The following figures show the recognized Devnagri characters displayed on computer screen.

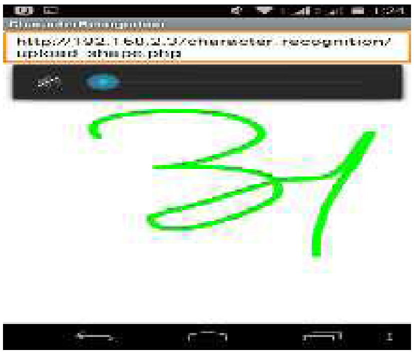

Fig 4 Handwritten Character on android aaav

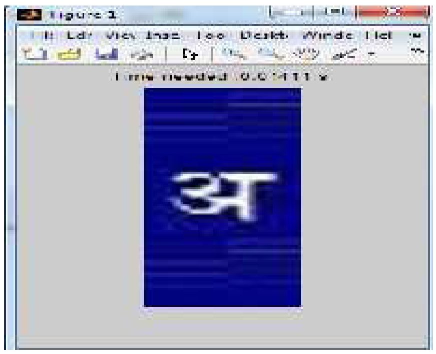

Fig 5 Recognized Character V

Fig 6 Handwritten Character on android Hk

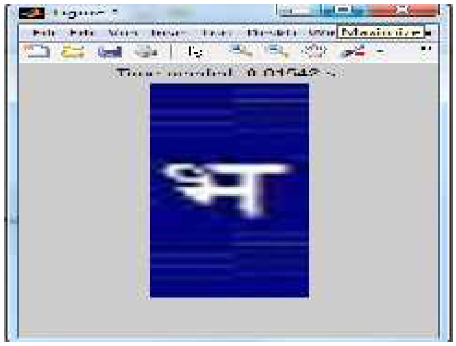

Fig 7 Recognized Character Hk

7. Conclusion

As such there is no recognition system exists for online recognition of handwritten characters with higher accuracy, we worked on almost all the characters and tried to develop a system with new technique particle swarm optimization. In this paper PSO and SVM for recognition of text from real time input using a combination of Android, PHP and MATLAB is discussed. The experimental results shown in Table 1, implies that PSO technique gives better accuracy as compared to SVM. The accuracy for the real time system developed in this work is around 90%.

Acknowledgements

We are very thankful to Sant Gadge Baba Amravati University (M.S) India for providing us a platform to do research work. We also acknowledge Jawaharlal Darda Institute of Engg & Tech Yavatmal,India for providing research lab to carry out our work. We are also thankful to H.V.P.M's College of Engg & Tech Amravati, India for their timely help.

References

1. Brijmohan Singh, Ankush Mittal "An Evaluation of Different Feature Extractors and Classifiers for Offline Handwritten Devnagari Character Recognition" *Journal of Pattern Recognition Research* 2 (2011) 269-277.

2. Srihari SN, Ball G "An assessment of arabic handwriting recognition technology", *CEDAR Technical Report, TR-03-07.* (2008)

3. Yin F, Liu C-L "Handwritten Chinese text line segmentation by clustering with distance metric learning", *Pattern Recognition* 42(12):3146–3157(2009)

4. C.Namrata Mahender "Structured based Feature Extraction of Handwritten Marathi Word", *International Journal of Computer Applications (0975 – 8887) Volume 16– No.6, February 2011*

5. C. Anton, C. tirbu ," Identify Handwriting Individually Using Feed Forward Neural Networks" *at International Journal of Intelligent Computing Research (IJICR),* Volume 1, Issue 4, December 2010

6. U. Pal, R. K. Roy, K. Roy, and F. Kimura, "Indian Multi Script Full Pin-code String Recognition for Postal Automation", *Proc. 10th Int. Conf. Document Analysis and Recognition, Barcelona, Spain,* Jul. 26-29, 2009, pp. 456-460.

7. B. B. Chaudhari, Digital Document Processing – "*Major Directions and Recent Advances", Springer,* London, 2007.

8. Simone Marinai and Hiromichi Fujisawa, "*Machine Learning in Document Analysis and Recognition",* Springer-Verlag, Berlin Heidelberg, 2008.

9. U. Pal, and B. B. Chaudhari, "Indian Script Character Recognition: a Survey", *Pattern Recognition,* vol. 37, 2004, pp. 1887-1899.

10. Utpal Garain, and B. B. Chaudhuri, "Segmentation of Touching Characters in Printed Devanagari and Bangla Scripts using Fuzzy Multifactorial Analysis", *IEEE Trans. Systems, Man and Cybernetics-Part C, Applications and Reviews,* vol. 32, no. 4, 2002, pp. 449-459.

11. Brijmohan Singh, Ankush Mittal "An Evaluation of Different Feature Extractors and Classifiersfor Offline Handwritten Devnagari Character Recognition" Journal of Pattern Recognition Research 2 (2011) 269-277.

12. Srihari SN, Ball G (2008) An assessment of arabic handwritingrecognition technology. CEDAR Technical Report, TR-03-07.

13. Yin F, Liu C-L (2009) Handwritten Chinese text line segmentationby clustering with distance metric learning. Pattern Recogn42(12):3146–3157

14. C.Namrata Mahender," Structured based Feature Extraction of HandwrittenMarathi Word",at *International Journal of Computer Applications (0975 – 8887)Volume 16– No.6, February 2011*

15. C. Anton, C. tirbu ," Identify Handwriting Individually Using Feed Forward Neural Networks" at International Journal of Intelligent Computing Research (IJICR), Volume 1, Issue 4, December 2010

16. U. Pal, R. K. Roy, K. Roy, and F. Kimura, "Indian Multi Script Full Pin-code String Recognition for PostalAutomation", Proc. 10th Int. Conf. Document Analysis and Recognition, Barcelona, Spain, Jul. 26-29, 2009,pp. 456-460.

17. B. B. Chaudhari, Digital Document Processing - Major Directions and Recent Advances, Springer, London,2007.

18. Simone Marinai, and Hiromichi Fujisawa, Machine Learning in Document Analysis and Recognition,Springer-Verlag, Berlin Heidelberg, 2008.

19. U. Pal, and B. B. Chaudhari, "Indian Script Character Recognition: a Survey", Pattern Recognition, vol. 37,2004, pp. 1887-1899.

20. Utpal Garain, and B. B. Chaudhuri, "Segmentation of Touching Characters in Printed Devanagari and BanglaScripts using Fuzzy Multifactorial Analysis", IEEE Trans. Systems, Man and Cybernetics-Part C,Applications and Reviews, vol. 32, no. 4, 2002, pp. 449-459.

21. Moraglio, A., C. Di Chio, J. Togelius, and R. Poli. 2008. "Geometric particle swarm optimization." Journal of Artificial Evolution and Applications ID 143624,

22. M. ALI ABED,"Simplifying Handwritten Characters Recognition Using a Particle Swarm Optimization Approach" at European Academic Research, Vol. I, Issue 5/ August 2013.