2009 International Conference on Advances in Recent Technologies in Communication and Computing

A Novel Technique for English Font Recognition Using Support Vector Machines

R.Ramanathan, L.Thaneshwaran, V.Viknesh, T.Arunkumar, P.Yuvaraj
Department of Electronics & Communication Engineering,
Amrita Vishwa Vidyapeetham,
Coimbatore, India
e-mail: r_ramanathan@ettimadai.amrita.edu

Dr. K.P.Soman
Center for Excellence in Computational Engineering and Networking (CEN)
Amrita Vishwa Vidyapeetham,
Coimbatore, India
e-mail: kp_soman@amrita.edu

***Abstract*—Font Recognition is one of the Challenging tasks in Optical Character Recognition. Most of the existing methods for font recognition make use of local typographical features and connected component analysis. In this paper, English font recognition is done based on global texture analysis. The main objective of this proposal is to employ support vector machines (SVM) in identifying various fonts. The feature vectors are extracted by making use of Gabor filters and the proposed SVM is trained using these features. The method is found to give superior performance over neural networks by avoiding local minima points. The SVM model is formulated tested and the results are presented in this paper. It is observed that this method is content independent and the SVM classifier shows an average accuracy of 93.54%.**

Keywords—English font recognition, Gabor filter, Support vector machine, Optical Character Recognition

I. INTRODUCTION

Font recognition is a fundamental issue in document analysis, which plays an important role in optical character recognition (OCR) and in reconstruction layout [6]. Though it is a time consuming approach it has a greater influence over automatic document processing (ADP). The information of a character's font is very important in the reconstruction of the text from the damaged or original layout and it is a significant factor in script identification. Apart from providing information about the font without any prior knowledge of the content, it helps to select the right character template while recognizing the characters, thus improving the character recognition rate.

Earlier proposed methods which use local typographical features[6] are proved to be less efficient compared to the recent global texture analysis methods such as weighted Euclidean distance (WED)[1], Apriori Optical Font Identification System (ApOFIS) [6] which works as a multivariate Bayesian classifier. Morris has examined the applicability of human vision models to typeface discrimination; he used Fourier amplitude spectra of images to extract global feature vectors used by a Bayesian classifier. Khoubyari and Hull [3] presented an algorithm that identifies the predominant font in a document [5]. Borji and Hamidi [1] examined the efficiency of WED algorithm in recognizing the Persian fonts.

In this paper, an algorithm based on Support Vector Machines (SVM) is proposed. The key point is using the texture analysis to extract the global features and using them to train the SVM.

II. BRIEF INTRODUCTION TO SVM

SVM is a tool used for classification [15] and is a recent addition to the toolbox of Artificial Intelligence. An SVM formulation can be easily solved using linear programming or quadratic programming. It is superior to neural networks as its formulation is in such a way that there is only one minima and it is the global minima. In a two-class problem SVMs construct a hyper plane, that separates the two classes along with making generalization error to be minimum. The two planes parallel to the classifier which passes through one or more points in the data set are called bounding planes which maximizes the margin between the two classes. Those points are called support vectors. In addition, SVM will achieve greater generalization ability. When a data is linear, classification is easy enough. If the data is non-linear, there arises the need of mapping the feature to higher dimensional space, which increases the computational complexity. Kernel methods exploit the convex optimization concept to achieve the maximal flexibility, generality, and performance in terms of both generalization and computational cost. The explicit mapping to higher dimensional space is avoided by using the Kernel trick. Thus, all computation will be done in input space itself.

A variant of SVM called the C-SVM has been used in this paper. Below given are the primal and dual formulations of a normal classification problem where the data is not linearly separable.
Primal:

$$\underset{w,\gamma,\xi}{Min} \; w^T w + C\varepsilon^T \varepsilon \tag{1}$$

Such that

$$D(AW - \gamma) + \varepsilon_i \geq 1 \tag{2}$$

978-0-7695-3845-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ARTCom.2009.89

IEEE computer society

Where D is a diagonal matrix, containing all the class values of the instances. A is the instance matrix. W is the weight matrix. γ Is the bias term and ε is the soft margin.
Dual:

$$\underset{u}{Min}\ \frac{1}{2}U^T D\left(K+\frac{I}{C}\right)DU - e^T U \qquad (3)$$

Such that

$$e^T DU = 0 \qquad (4)$$

$$U \geq 0 \qquad (5)$$

Where U is lagrangian multiplier. K is the kernel matrix. In dual, the ‘Kernel Trick’ is applied to map the data in to higher dimensional space where the instances are linearly separable implicitly. The kernel function used here is the Radial Basis Function (RBF) as it maps in to infinite dimensional space. It is defined as,

$$k(x,y) = \exp\left(-\sigma\left|x-y\right|^2\right) \qquad (6)$$

Where σ determines the slope of the function, x and y are the feature vectors in input space.

III. Methods Used For Feature Extraction

The efficiency of SVM mainly depends on the choice of features corresponding to the nature of the task to be performed. Here mean and standard deviation are selected as features. For the extraction of these features, two methods are used for pre-processing. They are as follows,
a. Otsu’s method
b. Gabor filters

A. Otsu’s method

Otsu's method is used to perform histogram shape-based image thresholding automatically, or, the reduction of a gray level image to a binary image [2]. The image to be thresholded is considered as image containing two classes of pixels (e.g. foreground and background) then calculates the optimum threshold separating those two classes which lies in the range [0,1] so that their combined spread (intra-class variance) is minimal. Any imaginary value in the image is neglected. It can also be further extended for multilevel thresholding called Multi Otsu method. Then the image is converted into binary image based on whether the intensity value of the pixel is greater or less than the calculated threshold. Some of the automated methods for finding threshold are

- Known Distribution
- Finding Peaks and Valleys
- Clustering(K-means variation)
- Clustering (The Otsu method)
- Mixture Modeling, etc

B. Gabor Filter

Then the part of the image needed is filtered with the help of Gabor filters since it is a better approximation to the receptive field profile of simple cells in visual cortex. It is defined by the following equation.

$$g_{\gamma,\eta,\phi,\lambda} = \exp(\frac{x'^2+\gamma 2 y'^2}{2\sigma^2}).\cos(\frac{2\pi x'}{\lambda}+\phi) \qquad (7)$$

$$x' = x\cos\theta - y\sin\theta \qquad (8)$$

$$y' = x\sin\theta - y\cos\theta \qquad (9)$$

The Gabor filters can be better used by varying the

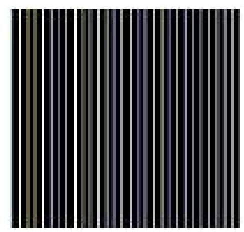

Figure 1. Time domain Gabor filter

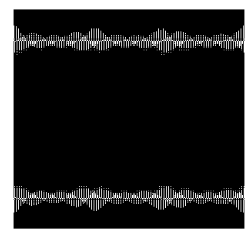

Figure 2. Frequency domain Gabor filter

parameters like λ, γ, φ and θ. In the above equations, x and y represent image coordinates; s is the standard deviation of Gaussian function which is usually set to 0.56 λ; λ is the wave length of cosine equation; γ characterizes the shape of Gaussian, circular shape for γ=1 and elliptic for γ<1 and θ represents the channel orientation and takes values in interval (0, 360). Since it is symmetric, θ varies from zero to 180. The response of this filter is nothing but the convolution given by the equation

$$\iint I(\varepsilon,\eta)g(x-\varepsilon, y-\eta)d\varepsilon d\eta \qquad (10)$$

The value of θ and σ must be taken under some considerations to make the choice of filter to be optimum.

IV. Building SVM Model

In the proposed system, the font recognition using SVM is carried through a strategy shown in the Figure 3.

A. Conversion to image files

The text files to be trained cannot directly be used for global feature extraction. Hence, they are converted to image files, which are further normalized and converted to binary image with the help of Otsu’s algorithm. The following steps can achieve it

- Compute the histogram and probabilities of each intensity level and set up initial class probabilities ($\omega_i(0)$) and class means ($\mu_i(0)$) .
- Step through all possible thresholds maximum intensity‘t’
- Update ωi and μi and Compute s=σ*σ and Desired threshold corresponds to the maximum ‘s’.
- Then the intensity of each pixel is compared with the threshold calculated.
- If the intensity is greater than the threshold it is considered as white else it is considered as black.

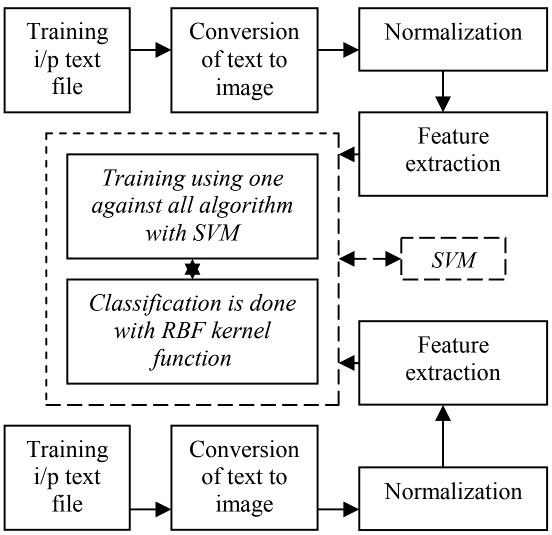

Figure 3. SVM Model

B. Feature Extraction

The binary image is passed through the Gabor channels. Three values (2.7, 4.1, and 5.4) are fixed for λ. The value for parameter θ is taken as kπ/8, where k is varied as 1, 2…8.Thus 24 Gabor channels are obtained. The binary

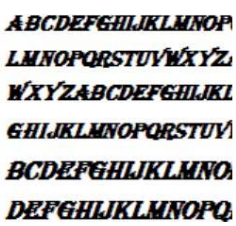

Figure 3. Training image

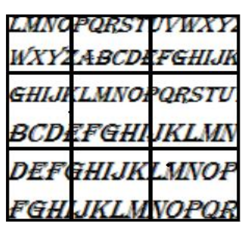

Figure 4. Processed image

image is divided into 9 non-overlapping blocks of equal size, for which 24 responses were found by passing through each channel. From these responses, mean and standard deviations are calculated. Another image is derived by taking the maximum of these filter responses per pixel. Thus, 50 features are extracted for each block. As a result, 9 instances are obtained for each image.

C. Learning process

Here six English fonts are taken for classification. Since, SVM cannot merely classify a document of various fonts. It is provided with a learning environment, using the training files of different styles (regular, bold, bold italic, italic) and is trained for classification by giving the features as inputs. Classification

The features are extracted from a given test file using the steps mentioned above. It is then fed to the trained SVM for classification. The developed SVM is evaluated and a 10-fold cross validation is done to verify the results. The accuracy of the SVM model is computed. The SVM model can be understood through the following steps:

- For each font, four images are chosen corresponding to each style.
- The image is resized to the dimension 300 x 300 and is subdivided into nine non-overlapping blocks.
- The features are extracted for each block using the above algorithm.
- Being a Multi class problem, this is solved using the one-against-all algorithm.
- The SVM Model is verified by 10-fold cross validation
- The influence of number of training instances over accuracy is also analyzed.

V. RESULTS AND DISCUSSION

A number of experiments for various trials have been carried out to evaluate the proposed algorithm. Six frequently used English fonts such as Times New Roman, Arial, Comic Sans MS, Courier New, Algerian, and Tahoma were combined with four styles regular, bold, italic and bold-italic were considered for developing the model. Initially 216 instances were taken for training and gradually incremented. The accuracy of the model for recognizing different fonts are computed over 10-fold cross validation and are tabulated in table1.It is noted that even with 216 instances the SVM model offers a good accuracy of around 95%. A slight variation in accuracy for larger instances results due to over learning. Accuracy results are shown graphically in Fig 6. Below

TABLE I. ACCURACY OF FONT RECOGNITION WITH 10-FOLD CROSS VALIDATION

Font	**Number of Training instances**			
	216	***432***	***648***	***864***
TimesNewRoman	92.12%	93.75%	89.68%	92.43%
Arial	92.12%	92.36%	91.43%	93.20%
Comic Sans MS	96.29%	96.75%	95.25%	96.60%
Courier New	95.83%	96.29%	94.90%	92.74%
Algeria	96.29%	91.43%	95.48%	93.82%
Tahoma	93.51%	91.43%	92.00%	92.59%

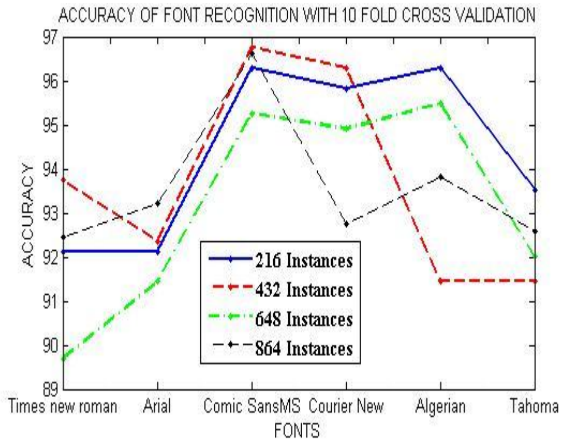

Figure 6. Accuracy of Font Recognition

The above figure (Figure 6) is zoomed to the scale from 89% to have better view of the accuracies and differences for various methods.

VI. CONCLUSION

A novel and efficient method of English font recognition is presented and discussed in this paper. The Gabor filters are effectively used for extracting the features used to train the proposed model. The model is developed as a content independent one with a good accuracy. The model is validated for different number of training instances. In addition, it is fine-tuned by changing the value of parameters Gamma and C in the SVM formulation to achieve the best results. The Radial Basis function Kernel is used for feature mapping. The proposed method is found to be an effective technique for font recognition and can be applied to any languages. Application of the technique to south Indian languages is under research.

REFERENCES

[1] Borji, and M. Hamidi. "Support Vector Machine for Persian Font Recognition" Proceedings of world academy of science, Engineering and technology Volume 22,July 2007,ISSN1307-6884.

[2] Xianpeng Lang, Feng Zhu,Yingming Hao Jinjun Ou. "Integral Image Based Fast Algorithm for Two-Dimensional Otsu Thresholding". *IEEE Computer Society Washington, DC, USA*

[3] S. Khoubyari, J. J. Hull, "Font and function word identification in document recognition", *Computer Vision and Image Understanding, Vol.63, No.1, pp.66-74, January 1996*

[4] Xiaoqing Ding, Tao Wu. "Character Independent Font Recognition on a Single Chinese Character". *IEEE Transactions on pattern analysis and machine intelligence, Vol.29, N0.2, February 2007.*

[5] Yong Zhu, Tieniu Tan, and Yunhong Wang. "Font Recognition Based on Global Texture Analysis" IEEE Transactions on pattern analysis and machine intelligence, Vol.23, N0.10, October 2001.

[6] Abdelwahab Zramdini and Rolf Ingold. "Optical Font Recognition Using Typographical Features" *IEEE Transactions on pattern analysis and machine intelligence, Vol.20, N0.8, August1998.*

[7] Fang yang, Xue-dong tian, Bao-lan guo. "An improved font recognition method based on texture analysis". Proceedings of the first International Conference on machine learning and cybernetics, Beijing, 4-5 November 2002.

[8] M.-C. Jung, Y.-C. Shin, and S.N. Srihari, "Multifont Classification Using Typographical Attributes," Proc. Sixth Int'l Conf. Document Analysis and Recognition, pp. 353-356, 1999.

[9] H. Shi and T. Pavlidis, "Font Recognition and Contextual

[10] A. Schreyer, P. Suda, and G. Maderlechner, ªFont Style Detection in Documents Using Textons,º Proc. Third Document Analysis Systems Workshop, Assoc. for Pattern Recognition Int'l, 1998.

[11] X. D. Tian, C. Zhang, "A Chinese document layout analysis method based on minimal spanning tree clustering", Proceedings of the 2nd International Conference on Machine Learning and Cybernetics, Xi'an, pp. 3183-3187, November 2003

[12] C. F. Lin, Y. F. Fang, Y. T. Juang, "Chinese text distinction and font identification by recognizing most frequently used characters", Image and Vision Computing, Vol.19, No.6, pp.329-338, April 2001.

[13] B. B. Chaudhuri, and U. Garain, "Automatic Detection of Italic, Boldand All-Capital words in Document Images", *Proc. of 14th InternationalConference on Pattern Recognition* - ICPR, Brisbane (Australia),pp.610-612, 1998.

[14] A. Bénédicte, H. Emptoz, "Font Type Extraction and CharacterPrototyping Using Gabor Filters". *International Conference on Document Analysis and Recognition*, p, 799-803, 2003.

[15] K.P.Soman, Loganathan.R, V.Ajay "Machine Learning with SVM and other Kernel Methods", Prentice Hall of India.