Identifying Handwritten Text in Mixed Documents

Conference Paper · January 2006

DOI: 10.1109/ICPR.2006.676 · Source: DBLP

Faisal Farooq
Siemens
24 PUBLICATIONS **487** CITATIONS
SEE PROFILE

Karthik Sridharan
University of Pennsylvania
69 PUBLICATIONS **2,289** CITATIONS
SEE PROFILE

Venu Govindaraju
University at Buffalo, The State University of New York
400 PUBLICATIONS **9,018** CITATIONS
SEE PROFILE

Identifying handwritten text in mixed documents

Faisal Farooq Karthik Sridharan Venu Govindaraju

CEDAR, University at Buffalo
Amherst, NY, USA, 14228
E-mail: {*ffarooq2,ks236,govind*}*@cedar.buffalo.edu*

Abstract

In this paper we present a system for classification of machine printed and handwritten text in mixed documents. The classification is performed at the word level. We propose a feature extraction algorithm for each word image based on Gabor filters followed by classification using an Expectation Maximization(EM) based probabilistic neural network that reduces overfitting of training data. An overall precision of 94.62% *was obtained for the Arabic script using the modified neural network. The accuracies obtained using a simple backpropagation neural network and an SVM were* 83.33% *and* 90.26% *respectively.*

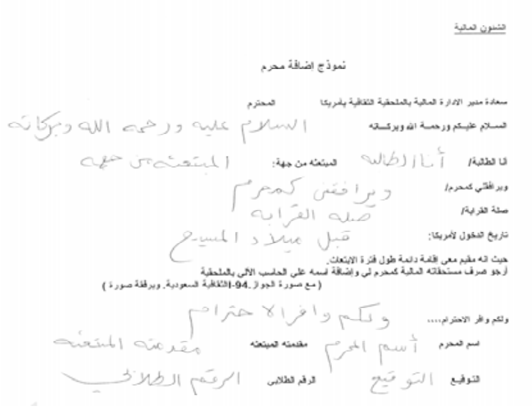
1 Introduction

The processing of document images prior to recognition plays a significant part in the development of Handwriting Recognition (HR) systems. In a document that has both machine print and handwritten text, it is important to distinguish between the two. We describe a method to identify handwritten words in a document image using Arabic as a representative script. This is because the task proves specially challenging in Arabic because the script is cursive in both machine print and handwriting. The accuracies achieved in this script can very well be translated to other scripts of similar nature.

In this paper we describe a method that extracts texture features from word images. An EM based neural network is used for classification to deal with the sparse training data that does not have representatives from all fonts and writing styles.

2 Previous Work

A neural network based classifier was suggested in [8] that used nine texture features to distinguish machine print from the handwritten text in bank checks. Srihari et al. [12] describe a block separation method where the classification is based on the frequency of the heights of the different components in the segmented block. It is assumed that a block with widely differing heights is handwritten and a block with uniform component heights is machine printed. A rule based approach was described by Pal and Chaudhari [11] for Devanagiri script. A similar approach was taken by Guo and Ma [6] by using projection profiles. These methods do not apply readily to other scripts. Zheng et al [14] proposed using a mix of run-length, crossing count, stroke orientation and texture features. Extracting all these features is a computationally expensive task and we believe that a minimal set of features is required for the actual task. Our hypothesis is that in handwriting, horizontal runs and gradients are not as uniform as in machine print. The advantage of our method is that it can be implemented at the word level as it captures the local structure of components in the document. A discrimination method that operates at



الشئون المالية

نموذج إضافة محرم

سعادة مدير الادارة المالية بالملحقية الثقافية بأمريكا المحترم

السلام عليكم ورحمة الله وبركاته السلام عليه ورحمه الله وبركاته

انا الطالبة/ أنا الطالبه المبتعثه من جهة: المبتعثه من جهه

ويرافقني كمحرم/ ويرافقني كمحرم

صلة القرابة/ صله القرابه

تاريخ الدخول لأمريكا: قبل ميلاد المسيح

حيث انه مقيم معي إقامة دائمة طول فترة الابتعاث.
أرجو صرف مستحقاته المالية كمحرم لي وإضافة اسمه على الحاسب الآلي بالملحقية
(مع صورة الجواز ,94-I الثقافية السعودية. ويرفقة صورة)

ولكم وافر الاحترام.... ولكم وافرالاحترام

اسم المحرم أسم المحرم مقدمته المبتعثه مقدمته المتعثه

التوقيع التوقيع الرقم الطلابي الرقم الطلابي

Figure 1. A sample document

0-7695-2521-0/06/$20.00 (c) 2006 IEEE

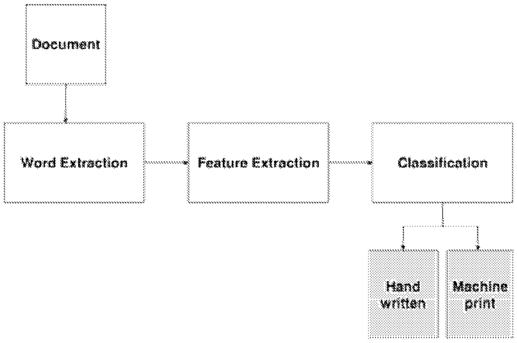


Figure 2. Components of the system

the word level was described in [4], using slope and stroke width histograms. However, the method was not trainable and thresholds were selected empirically.

Figure 2 shows a block diagram of our approach. It has 3 stages : (i) word extraction (ii) feature extraction and (iii) classification. In the word extraction stage, we binarize [10] the image and extract individual word images from the document [3]. Each word image is normalized by scaling to a fixed height while preserving the aspect ratio, hence the width of the word images vary. Directional Gabor filters are used to extract features from the word image. Classification is performed by a probabilistic neural network which is trained using an EM algorithm. This neural network combines solutions according to their posterior distribution to avoid overfitting based on the training data.

3 Feature Extraction

Gabor filters are directional filters that have been used for classification of textures and automatic script identification [13]. They have also been successfully used in address block location [7], logical labeling of document text blocks [1] and character prototyping [2]. Since direction of strokes and uniformity is a key feature, the use of Gabor filters seem to be ideally suited for the task.

Gabor functions are Gaussian functions modulated by a complex sinusoid. In $2D$, a Gabor function is given by:

$$h(x,y) = g(x',y').e^{2\pi jFx'}$$

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}e^{-\frac{1}{2}[(\frac{x}{\sigma_x})^2+(\frac{y}{\sigma_y})^2]}$$

where (x',y') are rotated components of (x,y),

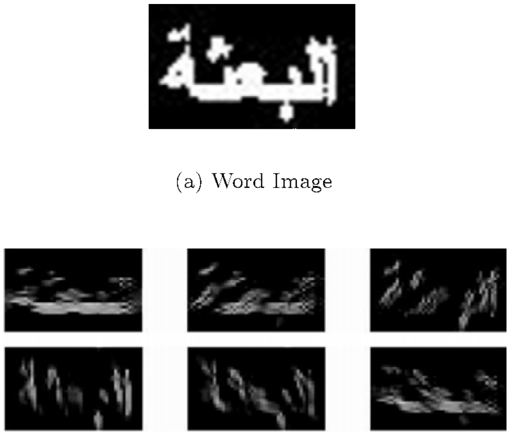
$$x' = xcos\theta + ysin\theta$$

$$y' = -xsin\theta + ycos\theta$$

and F is the radial frequency which for a given scale s is given by $F = F_0/s$. The output of the filter,

$$G_{\theta,s}(x,y) = \int I(s,t)h_{\theta,s}(x-s,y-t)dsdt$$

is an image with the components in the chosen direction becoming prominent. Since machine print has more uniformity as compared to handwriting and the same characters repeated in the text have strokes in the same direction, Gabor filters for feature extraction is a prudent choice. Figure 3(a) shows a sample word image extracted. Figure 3(b) shows the output of the Gabor filter for each direction at a single scale when applied to the word image in Figure 3(a).



(a) Word Image

(b) Output of Gabor filter

Figure 3. Extracting orientation information from six directions.

Since the word images all vary in their width the Gabor filter cannot be applied directly. For classifiers like neural networks or support vector machines (SVM) the feature vectors need to be of fixed size. This problem can be resolved by noting that the main information obtained from the Gabor filter output is the strength of the word image in each direction and scale which is given by the sum of the output of each filter resulting in a vector of size [number of scales × number of directions]. In order to make it font independent we normalize the output by dividing the sum of filter output by the sum of the output of an isotropic Gaussian filter,

$$Gauss(x,y) = \int I(s,t)g(x-s, y-t)dsdt$$

For direction θ and scale s

$$Gabor(\theta, s) = \frac{\int G_{\theta,s}(x,y)dxdy}{\int Gauss(x,y)dxdy}$$

In our implementation we use a set of 12 filters at 2 scales and 6 directions per scale. Thus for each word image we extract a 12–dimensional feature vector for classification.

4 Classification

The training set is generally sparse and does not cover all fonts. Traditional classifiers like SVMs and backpropagation neural networks tend to overfit sparse data. Figure 4 depicts the classification problem for identifying handwriting in mixed documents. As shown machine-print is distributed in clusters where as handwritten text is scattered in the feature space. The overfitting in a conventional classifier (straight line) leads to misclassification. Generalization (curved-dotted) is very important in such scenarios so that overfitting is avoided. This can be achieved by the Bayesian Neural Networks(BNN) [9] by integrating over the posterior distribution of the weights. That is, instead of finding one solution, many solutions are found and are weighted according to their posterior probabilities. The BNN outperforms many classifiers including the SVM. However BNNs need to sample high dimensional weight vectors. Markov Chain Monte-Carlo sampling methods, such as Langevian Monte Carlo method and Hamiltonian sampling methods can be used for the purpose. However these methods are computationally expensive.

A BNN for a binary classification can viewed as a linear combination of potential solutions according to their posterior probabilities. Since sampling is computationally intensive, we propose a new neural network where a layer of neurons use an error function which apart from penalizing neurons responsible for errors in classification, also penalizes neurons that are similar to each other. The idea is to make the neurons compete in finding different possible solutions. The part of the error function penalizing solutions that lead to misclassification is given by the sum of the square of the cosine of the neuron weight vector with respect to the weight vectors of the other neurons in the layer. A bias term is included in the weight vector to make sure that all the hyperplanes given by the neurons need not pass through the origin. Thus the error function of a single neuron is given by

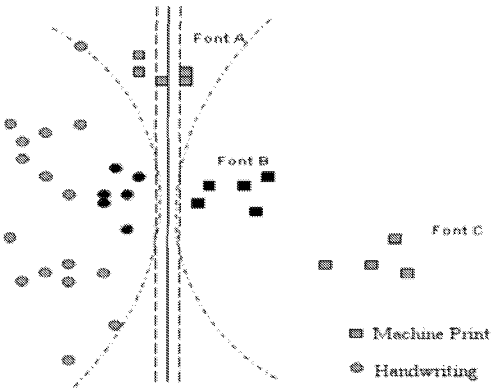


Figure 4. The classification task (*Black* - Training, *Grey* - Testing)

$$E_{i,k} = \frac{1}{2}(t_k - o_k)^2 + \frac{\beta}{2}\sum_{j!=i} \frac{w_i^T w_j}{|w_i||w_j|}^2 \qquad (1)$$

where

$$cos(w_i, w_j) = \frac{w_i^T w_j}{|w_i||w_j|}$$

and t_k is the target for the k^{th} instance and o_k is the weighted sum of the output of all the neurons according to their posterior probabilities. Therefore,

$$o_k = \sum_i P(w_i)o_{k,i}$$

The transfer function used is the classic sigmoid function. One way of looking at this error function is as the negative log likelihood of the posterior. Thus, we would be modeling the likelihood of the output to follow a Gaussian distribution with mean around the target and the prior to be a zero-mean Gaussian distribution of cosine similarity between the neuron weight and the other neurons. Zero mean of the cosine signifies that we are trying to model orthogonal neurons ($cos(90) = 0$). Parameter β decides the trade off between the error on classification and how "different" the solutions should be. By minimizing the error function we obtain weights that are as orthogonal (different) to each other as possible and yet classify well.

5 Performance Analysis

There is a lack of standard labeled handwritten datasets for training and testing purposes in Indic

Table 1. Performance analysis of the system.

	Back-Prop. Neural Net		SVM		EM Neural Net	
	Precision(%)	Recall(%)	Precision(%)	Recall(%)	Precision(%)	Recall(%)
Handwritten	62.26	95.19	74.26	97.12	94.68	85.58
Machine-print	97.83	79.02	98.82	87.76	94.93	98.25
Overall Performance(%)	83.33		90.26		94.62	

scripts [5]. We have collected handwriting samples from forms that have prompts in machine print. Figure 1 shows an example of the document. We collected 34 documents from 18 different writers. These were immigration forms in different font faces and styles. We used 5 documents for training purposes and the remaining for testing.

We measured the performance of our system by the precision and recall metrics, commonly used by the Information Retrieval (IR) community. Precision in our case would be the ratio of handwritten words labeled correctly to all words that are labeled as handwritten by our system. Recall is measured as the ratio of handwritten words labeled correctly to all handwritten words in the test set. Similarly the corresponding metrics for machine print are also calculated. Table 1 shows the summary of our experimental results. In order to evaluate the performance of our classification step, we compared the results by using a back-propagation neural network and an SVM for classification. The overall precision of our system is 94.62%. Our system outperformed a backpropagation neural network (83.33%) and also an SVM (90.26%).

6 Conclusion

Discrimination of handwritten and machine printed text is required in many document analysis and forensic applications. We have presented an algorithm for discriminating handwriting from machine print. The results have been shown for Arabic, however, our method is trainable and relies on the uniformity of strokes and curves in machine print compared to handwriting. Given the training data, our method can be adapted to other languages and scripts as well. Our method is robust even when large amounts of training data are not available.

References

[1] B. Allier, J. Duong, A. Gagneux, P. Mallet, and H. Emptoz. Texture feature characterization for logical pre-labeling. *Proc. Intl. Conference on Document Analysis and Recognition*, pages 567–571, 2003.

[2] B. Allier and H. Emptoz. Character prototyping in document images using gabor filters. *Proc. Intl. Conference on Image Processing*, pages 537–540, 2003.

[3] F. Farooq, V. Govindaraju, and M. Perrone. Pre-processing methods for arabic handwritten documents. *Proc. of the Intl. Conference on Document Analysis and Recognition*, pages 267–271, 2005.

[4] F. Farooq, V. Govindaraju, and M. Perrone. Processing of handwritten arabic documents. *Proc. of the 12th Conference of the Intl. Graphonomics Society*, pages 183–186, 2005.

[5] V. Govindaraju, S. Setlur, S. Khedekar, S. Kompalli, and F. Farooq. Enabling access to multilingual indic documents. *Workshop on Document Image Analysis for Libraries*, pages 122–133, 2004.

[6] J. K. Guo and M. Y. Ma. Separating handwritten material from machine printed text using hidden markov models. *Proc. Intl. Conference on Document Analysis and Recognition*, pages 439–443, 2001.

[7] A. Jain and S. Bhattacharjee. Address block location on envelopes using gabor filters. *Pattern Recognition*, 25(12):1459–1477, 1992.

[8] E. B. D. S. Jose, B. Dubuisson, and F. Bortolozzi. Distinguishing between handwritten and machine printed text in bank cheque images. *Document Analysis Systems*, 2423:58–61, 2002.

[9] R. M. Neal. *Bayesian Learning for Neural Networks.* Springer Verlag, 1996.

[10] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems Man and Cybernetics*, 9(1):62–66, 1979.

[11] V. Pal and B. B. Chaudhuri. Machine-printed and handwritten text lines identification. *Pattern Recognition Letters*, 22(3-4):431–441, 2001.

[12] P. W. Palumbo, S. N. Srihari, J. Soh, R. Sridhar, and V. Demjanenko. Postal address block location in real-time. *IEEE Computer*, pages 34–42, 1992.

[13] P. B. Pati, S. S. Raju, N. Pati, and A. G. Ramakrishnan. Gabor filters for document analysis in indian bilingual documents. *Proc. of Intl. Conference on Intelligent Sensing and Information Processing*, pages 123–126, 2004.

[14] Y. Zheng, H. Li, and D. Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE Transactions on PAMI*, 26(3):337–353, 2004.