CODEBOOK BASED HANDWRITTEN AND PRINTED ARABIC TEXT ZONE CLASSIFICATION

Jayant Kumar , jayant@umiacs.umd.edu

Department of Computer Science
University of Maryland College Park 20740 MD

ABSTRACT

In this work, we present a method for classifying handwritten and printed Arabic text zones in noisy document images. We use Three-Adjacent-Segment (TAS) [8] based features which capture properties of a script. We construct two different codebooks of the local shape features extracted from a set of handwritten and printed Arabic documents and use it to train both Support Vector Machine and Fisher's linear discriminant classifiers using normalized histograms. Due to robustness of TAS features to noise the proposed classification scheme is suitable for noisy document images where performance of other methods degrades drastically. Our experiments show that we can achieve 90–95% classification accuracy. This method is also robust to segmentation results which may contain segments at word, line or paragraph level.

Index Terms— document segmentation, document classification, two-class classification

1. INTRODUCTION

Although the community has made significant progress on the analysis and recognition of clean, structured documents, analyzing noisy documents with mixed content, remains a difficult task. This becomes more challenging when the content to be extracted has no predefined form and it can be present anywhere, and in any style on the page. It is common to find documents with handwritten text, logos, figures, pictures and background patterns along with traditional structured content. Handwriting on a document often indicates corrections, additions, or other supplemental information that should be treated differently from the main content [1]. Traditional methods for text extraction based on projection profiles and morphological operations fail to work on these kinds of documents. Methods based on texture features, such as [2], which do not assume any structure, work only when background pattern is not textured and document has no noise. The complexity of the problem is greatly increased by noise and the variability of handwriting [3].

Analysis and recognition of documents containing Arabic script has not received as much attention as other scripts in spite of the fact that Arabic characters serve as scripts for several languages such as Arabic, Farsi, Urdu and Uygur [4], covering more than thirty countries. Until recently, much less work has been done on the segmentation and recognition of handwritten Arabic text and even less on noisy Arabic. Due to the unique nature of the script, existing methods do not always prove to be the most effective. Most Arabic character recognition systems, assume the clean documents and start by segmenting the text sequentially at line, word and/or character levels. Zahour et al. [5] proposed a method based on horizontal projections to first extract text blocks before text line segmentation in historical arabic documents. They developed a new segmentation method [6] suited for Arabic historical manuscripts, to segment the document image into three classes: text, graphics and background. Recently, Faisal et al. [7] discussed preprocessing methods for handwritten Arabic and proposed a method for baseline detection of Arabic words.

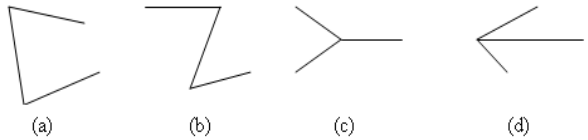


Fig. 1. Basic TAS types (a) C-Type (b) Z-Type (c) Y-type (d) ←-type

In this report we present a method for classifying handwritten and printed Arabic text regions in noisy monochromatic documents. This is a prerequisite step before applying any textline or word segmentation. It is assumed that these regions are already segmented from the image and given as input to our classification method. We use features extracted from local shape properties of the script, which are invariant to scale and rotation. Triple-Adjacent-Segment (TAS) features, a special case of kAS (k-Adjacent Segments) were first introduced in [8] and have been shown to be reliable in capturing local shape properties of a given object and successfully used for object detection [9] and script identification [10]. In [11] , it was shown that detection of only handwritten Arabic zones in a document with mixed content can be modeled as a

one class classification using TAS-features. But in presence of printed zones their method suffers from high false positives due to similar nature of printed and handwritten Arabic script. We extend their work by using two codebooks instead of one and focus on only two class classification. Our experiments are based on the assumption that variation between similar TAS features will be relatively less in printed text as compared to handwritten text. This is a reasonable assumption because one would expect TAS of same type to be more repeatable in case of printed and less repeatable in handwritten. We try to capture this variability using features obtained from two codebooks.

In Section 2 we explain the steps involved in extracting TAS features and constructing the shape codebook. We explain in detail our experiments and review results in Section 3. Finally we conclude the paper in Section 4 with some pointers to future work.

2. TEXT ZONE CLASSIFICATION

The purpose of zone classification is to label each segmented zone as one of a set of predefined types such as text, images, graphics and tables [13]. First, we obtain all the zones present in the document using an improved zone segmentation method based on the Vornoi segmentation [14]. Then we manually select a subset of representative printed and handwritten text zones for creating a shape codebook for both printed and handwritten Arabic. Finally, we train a two-class classifier using the features of the distribution of codewords in the codebooks. Figure 2(a) shows a sample document image from our dataset with segmented zones.

2.1. Constructing A Shape Codebook

2.1.1. Feature Extraction

Using Canny edge detector [15], we first obtain a list of edges present in the image. Then we find a similar list of line segments by fitting a line to each edge segment with a specified tolerance. We then group neighboring segments accordingly in the underlying connected components (CC). Every triplet within each CC forms one of the four basic TAS types defined as shown in Figure 1. For each extracted TAS, we determine its TAS-type along with the length and orientation features of its segments. Figure 2(b) depicts the TAS contours of a given text zone. Figure 3 shows a typical handwritten and printed zone obtained after the segmentation.

2.1.2. Obtaining the TAS Codewords

In order to construct an indexed shape codebook, we compute the symmetric dissimilarity between two TAS features T_a and T_b as shown in Equation 1

$$D(T_a, T_b) = w_\theta \sum_{i=1}^{3} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{3} \log\left(|l_i^a / l_i^b|\right) \quad (1)$$

where l_i^x and θ_i^x are the length and orientation of line segment i in TAS feature T_x, respectively. $D_\theta \in [0, 1]$ is the difference in orientation normalized by $\pi/2$ and w_θ is empirically set to 2 to emphasize the difference in orientations more than length as the later may be less accurate due to fitting.

For each given zone, the pairwise dissimilarity is computed for all pairs of TAS features. Normalized Cuts [19] is used to formulate the feature clustering as a graph partitioning problem and cluster the TAS features using the pairwise dissimilarities. The weight $w(T_a, T_b)$ on an edge connecting two nodes T_a and T_b in the graph is defined as a function of their distance, as shown in Equation 2

$$w(T_a, T_b) = \exp\left(-\frac{D(T_a, T_b)^2}{\sigma_D^2}\right) \quad (2)$$

where σ_D is a scaling parameter. The exact solution to Normalized Cuts is NP-complete. We use a fast implementation explained in [19] for finding the discrete near-global optima, for faster convergence. In each cluster, we select an exemplary codeword which is the TAS instance closest to the center of cluster. In addition, each exemplary codeword is associated with a cluster radius, which is defined as the maximum distance from the cluster center to all the other TASs within the cluster. The final codebook C is composed of all exemplary TAS codewords. Through clustering, translated, scaled and rotated versions of TAS feature types are grouped together. Figure 4 shows the results of TAS clustering using Normalized Cuts.

(a) (b)

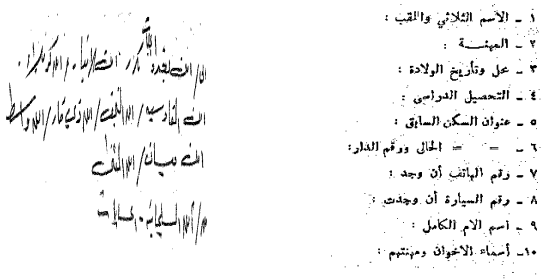
Fig. 2. (a) Sample image from our dataset with segmented zones (b) A text zone and corresponding TAS contours

2.2. Computing the Zone Descriptor

For each segmented zone, we construct a descriptor that provides statistics of the frequency of each TAS feature occurrence. For each detected TAS feature we increment the number of occurrence of the entry which is nearest to it. We do so only when the distance between this TAS feature and the nearest entry is less than the corresponding cluster radius.

$$D(T_a, C_k) < r_k \tag{3}$$

where r_k is the radius of the cluster for which C_k is the exemplar. We combine the two histograms obtained using printed and handwritten codebooks to obtain a single feature vector for each zone.



(a) (b)

Fig. 3. A typical (a) handwritten zone and (b) printed zone

(a) TAS type 1

(b) TAS type 2

(c) TAS type 3

Fig. 4. Ten most frequent exemplary codewords of each TAS type in handwritten zones.

2.3. Two-Class Classification

We used a freely available Matlab toolbox for pattern recognition called PRTools [20] for training two-class classifiers for zone classification. The classifiers available in PRTools can be broadly put in three categories : Linear and high degree polynomial classifiers, Normal Density based classifiers and Nonlinear classifiers. We chose two classifiers one from each linear and nonlinear category for our purpose : Fisher's least square linear Classifier (FISHERC) and Support Vector Classifier (SVC). FISHERC [17] finds the linear discriminant function between the classes in the dataset by minimizing the errors in least square sense whereas a nonlinear Support Vector Machine (SVM) [18] gives a decision function $f(\mathbf{x}) = sign(g(\mathbf{x}))$ as explained by equation 4:

$$f(\mathbf{x}) = \begin{cases} class1 & \text{if } g(\mathbf{x}) = +1 \\ class2 & \text{if } g(\mathbf{x}) = -1 \end{cases} \tag{4}$$

for an input vector **x** where

$$g(\mathbf{x}) = \sum_{i=1}^{l} w_i K(\mathbf{x}, \mathbf{z}_i) + b \tag{5}$$

$\mathbf{z}_i$ are representatives of training examples called support vectors. $K(\mathbf{x}, \mathbf{z}_i)$ is a kernel that implicitly maps vectors to a higher dimensional space.

Table 1. Number of clusters in codebook

Codebook	Type1	Type2	Type3	Total
1	30	30	10	70
2	20	10	0	30

Table 2. Total number of TAS features in codebook

	Type1	Type2	Type3
Handwritten	4523	2017	975
Printed	4225	1960	817

Table 3. Classification results using Type 1 codebooks

Classifier		HWZ	PTZ	Accuracy
SVC	Train	129	97	
	Test	32	21	86.9%
FISHERC	Train	129	97	
	Test	32	21	79.24%

Table 4. Classification results using Type 2 codebooks

Classifier		HWZ	PTZ	Accuracy
SVC	Train	129	97	
	Test	32	21	92.45%
FISHERC	Train	129	97	
	Test	32	21	83.01%

Table 5. Results using reduced cluster size

Classifier		HWZ	PTZ	Accuracy
SVC	Train	129	97	
	Test	32	21	90.5%
FISHERC	Train	129	97	
	Test	32	21	79.2%

3. EXPERIMENTAL RESULTS

Our dataset contains 100 monochromatic document images which contains both printed and handwritten Arabic text. After segmentation and basic filtering we obtained 118 printed and 161 handwritten zones. We observed that the frequency of occurrence of TAS type 4 ($\leftarrow$ type) was relatively lower in handwritten and printed Arabic text and hence we used only first three TAS-types for feature computation and classification. Figure 4 shows ten most frequent codewords of each TAS type. We also used different number of clusters for each TAS type depending on the number of features of corresponding type. We experimented with two different sizes of codebook to see the quantization effect on classification. Table 1 shows the number of clusters of each TAS type in two different codebooks created. We divided our data into training and test sets to obtain an average performance of the classifier. For classification we used the Matlab Pattern Recognition toolbox [20]. Table 2 provides the total number of features extracted for creating the codebook for handwritten and printed Arabic text. Table 3 and 4 provides the results obtained for classification using FISHERC and SVC classifiers with different sizes of codebook as specified in Table 1. For SVC we used polynomial proximity mapping (kernel) function.

Table 5 shows the results when the histogram compuation was done by reducing the cluster size to half. This was done to test the assumption that on an average the new TAS features will be close to the center of clusters in case of printed text. The first column in Table 3, Table 4 and Table 5 specifies the name of classifier used. Second and third column specify the number of handwritten text zones (HWZ) and printed text zones (PTZ) used in training and testing. Fourth column assigns the percentage of correct classification.

4. CONCLUSION

We experimented with TAS based features obtained from two different codebooks for classifying handwritten and printed text regions in noisy document images. The main advantage of using TAS features is that it is robust to background noise. Also normalized histogram based features used in our experiment are robust to size of a zone. Our results with two different classifiers show that features obtained using two codebooks are more effective than just one for classifying handwritten and printed Arabic text. At the same time, we did not observe any improvement by changing the threshold for the feature computation at lower cluster sizes. Also we can conclude from our experiments that having a codebook of smaller size allow us to capture more within-class consistency in printed case. In future, we plan to combine multiple classifiers to further improve the classification results, especially the outlier acceptance error. Also, we will use clustering techniques that automatically finds the optimal number of clusters for each TAS type. Features based on similarity of TAS within a zone can provide further improvement.

5. REFERENCES

[1] Yefeng Zheng, Huiping Li and David Doermann, “Machine Printed Text and Handwriting Identification in Noisy Document Images”, IEEE Trans. Pattern Anal. Mach. Intell., 26(3), pp. 337-353, March 2004.

[2] D. Chetverikov, J. Liang, J. Komuves, and R. Haralick, “Zone Classification Using Texture Features”, Proc. 13th Intl Conf. Pattern Recognition, pp. 676-680, Vienna, 1996.

[3] Abuhaiba I.S.I., Mahmoud S.A., Green R.J., “Recognition of handwritten cursive Arabic characters”, IEEE Trans. Pattern Anal. Mach. Intell., Vol 16, pp. 664-672, June 1994.

[4] Amin A. “Off-line Arabic character recognition : The state of the art”, Pattern Recognition, Vol. 31, pp. 517-530, 1998.

[5] A. Zahour , L. Likforman-Sulem , W. Boussellaa , B. Taconet, “Text Line Segmentation of Historical Arabic Documents”, Proc. Int’l Conf. Document Analysis and Recognition, Vol 1, pp. 138-142, September 23-26, 2007.

[6] W. Boussellaa, A. Zahour, B. Taconet, A.Benabdelhafid, A. Alimi, “Segmentation texte /graphique : Application au manuscrits Arabes Anciens.”, Neuvime Colloque International Francophone sur lEcrit et le Document, Fribourg, Suisse, 18-21 Septembre 2006,pp. 139- 144.

[7] F. Farooq, V. Govindaraju, and M. Perrone, “Preprocessing Methods for Handwritten Arabic Documents”, Proc.

Int'l Conf. Document Analysis and Recognition, pp. 267-271, 2005.

[8] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid, "Groups of adjacent contour segments for object detection", Technical Report, 2006.

[9] Xiaodong Yu, Yi Li, Cornelia Fermuller and David Doermann, "Object Detection Using Shape Codebook", British Machine Vision Conference, December 2007.

[10] Guangyu Zhu, Xiaodong Yu, Yi Li and David Doermann, "Unconstrained Language Identification Using A Shape Codebook", Int'l Conf. on Frontiers in Handwritting Recognition, pp. 13-18, 2008.

[11] Jayant Kumar, W. Abd-Almageed and D. Doermann. Handwritten Arabic Text Zone Detection using A Shape Codebook. Submitted to Intl. Conf. on Image Processing (ICIP 09), 2009.

[12] W. Abd-Almageed, M. Agrawal, W. Seo and D. Doermann. "Document Zone Classification Using Partial Least Squares and Hybrid Classifiers", International Conference on Pattern Recognition, 2008.

[13] Y. Wang, I. T. Phillips, and R. M. Haralick, "Document zone content classication and its performance evaluation", Pattern Recognition, 39(1):5773, 2006.

[14] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram", Comput. Vis. Image Underst., 70(3):370-382, 1998.

[15] J. Canny., "A computational approach to edge detection", IEEE Trans. Pattern Anal. Mach. Intell., 8(6):679-697, 1986.

[16] Bishop, C. (1995), "Neural Networks for Pattern Recognition", Oxford University Press, Walton Street, Oxford OX2 6DP.

[17] Raudys, Ł. and Duin, R. P. 1998. Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. Pattern Recogn. Lett. 19, 5-6 (Apr. 1998), 385-392.

[18] Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, 1995

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation", IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888-905, 2000.

[20] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, S. Verzakov PRTools4.1, A Matlab Toolbox for Pattern Recognition, Delft University of Technology, 2007