2011 International Conference on Document Analysis and Recognition

Identification of Indic Scripts on Torn-Documents

Sukalpa Chanda
Dept.of Computer Science and Media Technology
Gjøvik University College
Gjøvik-2815, Norway
E-mail:- sukalpa@ieee.org

Katrin Franke
Dept.of Computer Science and Media Technology
Gjøvik University College
Gjøvik-2815, Norway
E-mail:- kyfranke@ieee.org

Umapada Pal
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
Kolkata-700108, India
E-mail:- umapada@isical.ac.in

***Abstract*—Questioned Document Examination processes often encompass analysis of torn documents. To aid a forensic expert, automatic classification of content type in torn documents might be useful. This helps a forensic expert to sort out similar document fragments from a pile of torn documents. One parameter of similarity could be the script of the text. In this article we propose a method to identify the script in document fragments. Torn documents are normally characterized by text with arbitrary orientation. We use Zernike moment–based feature that is rotation invariant together with Support Vector Machine (SVM) to classify the script type. Subsequently gradient features are used for comparative analysis of results between rotation dependent and rotation invariant feature type. We achieved an overall script-identification accuracy of 81.39% when dealing with 11 different scripts at character/connected-component level and 94.65% at word level.**

Keywords- Script Identification; Torn Document; Gaussian Kernel SVM; Computational Forensics.

I. INTRODUCTION

Questioned-document examination process often requires to analyze a heap of torn documents. In such cases an automated system can sort out similar document fragments, and narrow down the search space of a forensic expert. A notion of similarity between two document fragments could be the script present in those two documents. Script Identification technique can be used to sort out similar document fragments which might come from the same source. This can be used as a criterion for linking two or more different document fragments to a document page and/or same source of origin. Lot of research has been done on script identification already. Yet the present state of the art is in-sufficient to address the challenges of script identification in context of document fragments. The adversaries involved in script identification on torn documents are as follows: (i) Scarcity of text/data content. Please note that all images of Fig.1 consist of very few text/words. (ii) Multiple orientation of text. (iii) Arbitrary background type for document fragments. In this article we intend to propose a script identification scheme based on Zernike moments/Gradient features for torn document fragments, which could be used as a part of an automatic questioned document examination system in context of Indian scripts. A brief review of some published research work on script identification is given in the following paragraph.

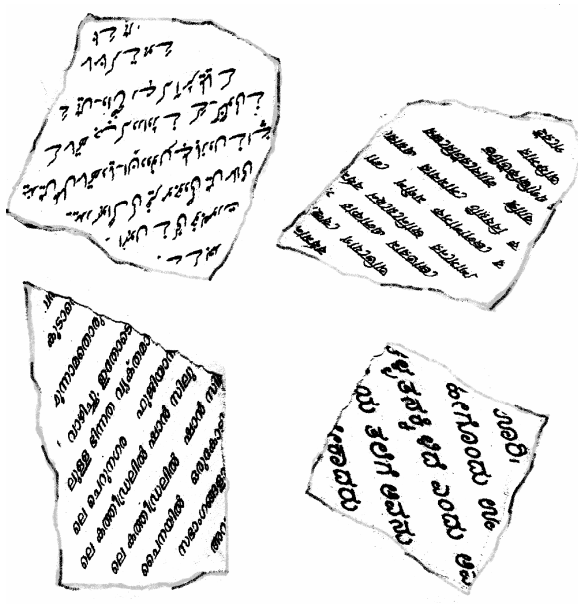


Figure 1.Torn documents consisting of text in some Indic scripts in multiple orientations.

Among the pieces of earlier work on script identification, Spitz [2] developed a method to separate Han-based or Latin-based script. He used optical density distribution of characters and frequently occurring word shape characteristics. Jaeger, Ma, and Doermann [14] used K-NN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean and Hindi scripts using cluster-based templates. An automatic script identification technique has been described by Hochberg, Kelly, Thomas and Kerns [7]. Using fractal-based texture features, Tan [3] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text. All the above mentioned works deal with non-Indian script. Among Indian script, there are some related works. Pal, Sinha and Chaudhuri [4] proposed a generalized scheme for line-wise script identification from a single document containing twelve Indian scripts. Sinha, Pal and Chaudhuri [13] narrated a word-wise script identification scheme with a combination of Indian languages. Dhanya and Ramakrishnan [5] mentioned a Gabor-filter-based technique for word-wise segmentation from bi-lingual

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.149

documents containing English and Tamil scripts; they have used classifiers like LSVM and K-NN. Patil and Subareddy [8] proposed a neural-network-based system for word-wise identification of English, Hindi and Kannada language scripts. Zhou, Lu and Tan [6] proposed a Bengali/English script-identification scheme using connected component analysis. In this paper a technique for script identification in torn documents is proposed that consist of Roman and all major Indic scripts. In the proposed method a study of two different feature types are conducted. A comparative analysis between a rotation invariant feature type and rotation dependent feature type is performed. For both cases, feature extraction is performed on each connected component/character-component found in a document fragment. Later the extracted features were passed to a Support Vector Machine (SVM) classifier. Classification of a document fragment (words) to a particular script is done based on majority voting of each recognized character component of the document. Classification results are reported at three different levels: (i) connected component/character-component level (ii) group of character-components /word level (iii) entire document fragment level.

II. METHODOLOGY

Script identification for the whole torn document is dependent on a hierarchy based classification. At the bottom most level of this hierarchy is the connected-component/character-components. In the middle level are words, which are formed by groups of character-component/connected-component. At the highest level the whole torn document which is formed by a collection of words. Classification of connected/character-component present in a torn document is done first. Based on majority voting of script amongst character-component present in a word, we decide the script of the word. Finally considering majority voting amongst script of all words present in the torn document, we decide the script type of the document. We noticed that arbitary orientation of text makes it difficult to define a 'word' for most of the script. In few scripts like Devnagari and Bengali the presence of a Head-line connects all characters together (See Fig.2 for illustration). In such cases defining a word is easy due to the presence of head-line even with arbitrary orientation. But for Roman and some other Indic-scripts like 'Oriya', 'Tamil', 'Telegu' etc, characters sit beside each other in same horizontal line and form a word. In an arbitary orientation scenario this won't be possible as they are not always in a horizontal line. For e.g. consider the Oriya word in Fig.2. As a consequence we need to deploy some pre-processing techniques, which will help us to define a group of character-components as a 'word'. One might argue that we can directly apply any rotation invariant feature extraction scheme on the character-components, and based on majority voting on all character-components present in a torn document, we can decide the script type for the document. This won't work always in Indian subcontinent scenario. Being a multilingual country it's very common in India to have multiple scripts in a single document page. It is quite possible that two scripts simultaneously occur in a document fragment of that document page. As a result we also need to identify the script at word level in a torn document. Then it can be used for some questioned document examination process in Indian sub-continent.

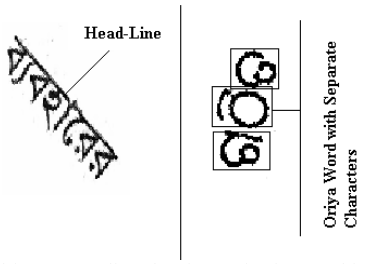
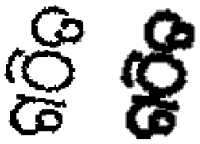


Figure 2. Word in a Bengali and Oriya script in an arbitrary orientation scenario.

To form word boundary in case of arbitrary oriented character-components for scripts like 'Oriya', 'Tamil', 'Telegu' etc, we used a mathematical morphology based dilation operator. Details of our method are narrated in Section III.

III. PRE-PROCESSING

All input grayscale images are converted to their binary equivalent. Next we perform the following tasks: (i) Perform character-component segmentation and bloating. (This is done with the help of *'region growing'*, applying five-times-dilation operation of Mathematical Morphology. The structuring element for the morphology is of size 7×7. See figure 3 for illustration of region growing). (ii) It can be noted that the original input image consist of a word of three separate characters, using our region growing technique we fused them to get a word boundary. (iii) Perform connected component analysis and word-labeling. (iv) For each labeled word component determine its direction of highest variation (extension) by implementing principal-component analysis (PCA) discussed in [16]. (v) Rotate each word component according to the direction of its first eigenvector. (Task iv and v are done solely to implement our rotation dependent feature extraction method).



(a) (b)

Figure 3. (a) Oriya word with separate character, (b) corresponding output after applying region growing method.

A. *Pre-processing and working methodology for Zernike moment feature*

Let B_W and B_C are the same copies of the binary image. In one binary image 'B_W', Region growing operation is performed on all connected components present in the image 'B_W'. As a result characters close to each other fuse

and form a much bigger connected component which we can term as 'word-component' (See Fig.3b). Now a component labeling in 'B_W' gives us our desired word boundary in 'B_W'. Other copy of the Binary image 'B_C' is not processed by region growing operation. We map the word boundaries obtained from 'B_W' on the image 'B_C'. We also calculate the average length ($A._L$) and width ($A._W$) of all character-component found in 'B_C'. Considering the word boundary we do component labeling inside a binary image 'B_C', (to get individual character-component present within the word boundary. Then each character–component with length >= $A._L$ and width >= $A._W$ are normalized to a square matrix. The size of this normalized matrix is considered with respect to *Max* ($A._L$,$A._W$). Zernike moment-based features are extracted from each of those size-normalized character-components and passed to a classifier. Classifier decides the script type for each character-component of the word.

B. *Pre-processing and working methodology for Gradient feature*

Gradient features are not rotation invariant as Zernike – moment-based features. So for extracting gradient features from text, we need to fix the orientation of the text. As mentioned earlier, after region growing operation characters in a word generally touch each other as shown in Fig.3b and a word become a single connected component (word-component). We perform word-component labeling in 'B_W'. Inside the word component in 'B_W', we calculate the distribution of black pixels. A PCA-based method is deployed to detect the orientation of the word. Details about it can be found in [17]. Word-component labeling in 'B_W' helps us to get word boundary from the binary image 'B_C' which is not processed by region growing operation. We copy the word area from the 'B_C' and fix the orientation of the word. After orientation of the word is fixed, a component labeling is performed within this word image. Individual character-components are processed for gradient feature extraction and extracted feature vector is passed on to the classifier. Classifier decides the script type for each character-component found in the word.

IV. Feature Extraction

Considering the possible arbitrary orientation of text present in a torn document, we looked for a feature extraction scheme which is rotation and scale invariant. As a consequence we initially experimented with various moment-based features like Hu, Zernike etc. We got best results with Zernike and we made all further experiments using Zernike moments-based features. Even though we got encouraging results with Zernike moments, we were curious to compare its efficacy in comparison to a rotation dependent feature extraction method. We had prior experience of fixing orientation of text present in such torn documents [17]. We used a similar morphology and PCA-based approach to deduce the orientation of the text, thereafter rotating the piece of text to normal orientation we extracted gradient-based features. In the following two sub-sections we will narrate the feature extraction methodology used to obtain our Zernike moment-based features and Gradient features.

A. *Zernike Moment Feature*

Zernike moment features are rotation invariant in nature. Two dimensional Zernike moment can be computed using the formula:

$$A_{mn} = \frac{m+1}{\pi} \int_x \int_y f(x,y)[V_{mn}(x,y)] \; * dxdy$$

where $x^2 + y^2 \le 1$ and $m - |n| = even, |n| \le m$

Here m = 0, 1, 2… ∞ defines the order and f(x, y) is the function being described and * denotes the complex conjugate. n is an integer implying the angular dependence.

For a discrete image pixel *P(x,y)*, the integrals are changed to summation, and the above equation gets transformed to the following:

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y P(x,y)[V_{mn}(x,y)] \; *$$

where $x^2 + y^2 \le 1$.

For our case the idea is to map the image of the size-normalized character-components to the unit disc using polar coordinates, where the centre of the image is the origin of the unit disc. Those pixels falling outside the unit disc are not used in our computation. We got best results with m=13 when basis functions with negative repetition are included, and it gives us a feature vector of dimension 105. Details about the feature can be found in [18].

B. *Gradient Feature*

To obtain 400-dimensional gradient features [10] we apply the following steps. (i) The input binary image is converted into a gray-scale image applying a 2 × 2 mean filtering *5* times. (ii) The resultant gray-scale image is then normalized. (iii) Normalized image is now segmented into 9×9 blocks. (iv) A Roberts filter is applied on the image to obtain gradient image. The arc tangent of the gradient (direction of gradient) is quantized into 16 directions and the strength of the gradient is accumulated with each of the quantized direction. By strength of Gradient ($f(x,y)$) we mean $f(x,y) = \sqrt{(\Delta u)^2 + (\Delta v)^2}$ and by direction of gradient ($\theta(x,y)$) we mean $\theta(x,y) = \tan^{-1} \frac{\Delta v}{\Delta u}$,

where $\Delta u = g(x+1, y+1) - g(x,y)$, and

$\Delta v = g(x+1, y) - g(x, y+1)$ and $g(x,y)$ is a gray scale at (x, y) point. (v) Histograms of the values of 16 quantized directions are computed in each of 9×9 blocks. (vi) 9×9 blocks are finally down sampled into 5×5 by a Gaussian filter. Thus, we get 5× 5×16 = 400-dimensional feature.

V. Classifier

In our experiments, we have used a Support Vector Machine (SVM) as classifier. The SVM is defined for two-class problem and it looks for the optimal hyper plane, which maximizes the distance, the *margin*,

between the nearest examples of both classes, named support *vectors* (SVs). Given a training database of M data: $\{x_m | m=1,...,M\}$, the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b$$

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem [11]. The linear SVM can be extended to various non-linear form, and details can be found in [11] [12]. In our experiments we noted Gaussian kernel SVM outperformed other non-linear SVM kernels, hence we are reporting our recognition results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$[k(x,y) = \exp(-\frac{\| x-y \|^2}{2\sigma^2})].$$

For Zernike moment-based features, Gaussian Kernel gave highest accuracy when $(1/2\sigma^2)$ is set to *0.00006* and penalty multiplier is 3. Due to such low value of $(1/2\sigma^2)$ we can conclude that the Zernike moment-based features generated for our 11 class problem makes classification more linear in nature. For gradient features we noticed that Gaussian kernel gave highest accuracy when $(1/2\sigma^2)$ is set to *36.00* and the penalty multiplier is 3. The high value of $(1/2\sigma^2)$ for classification with gradient feature indicates that more non-linearity is involved in classification task with gradient feature.

VI. Experimental Setup, Results and Discussions

We evaluated efficacy of both feature types separately. Classification accuracy for both feature types are recorded for all three following levels:(i) character-component (Level I) (ii) word/group of character-component, (Level II), and (iii) entire document fragment, (Level III). A brief on our dataset can be found in sub-section 'A'. In sub-section 'B' we present the accuracy for all 11 class of scripts (Bengali { class-1 }, Devnagari { class-2 }, Roman { class-3 }, Oriya { class-4 }, Gurumukhi { class-5 }, Gujarati { class-6 }, Telegu { class-7 }, Tamil { class-8 } Kannada { class-9 }, Malayalam { class-10 } and Urdu { class-11 }) at character/connected-component level in a graphical format. The accuracies on other two levels (word and document) are reported in two sub-sections 'C' and 'D' respectively. Later in sub-section 'E' we also present a confidence score distribution for both types of feature. By confidence score value of recognition we mean the probability estimation of the recognized class [9].

A. Dataset Details

To the best of our knowledge, there is no publicly available database suitable for our defined problem (torn documents with Indic scripts). We developed our own dataset to evaluate our proposed method. Utmost care is taken to ensure the presence of adversaries normally found in any torn document. Training dataset consist of 112 torn documents. The test dataset consists of 130 torn documents. From training and test dataset we obtained 7281 and 8130 connected/character-component, respectively. We considered 11 different scripts comprising of Bengali (Bangla), Devnagari, Oriya, Urdu, Malayalam, Gujarati, Telegu, Tamil, Kannada, Gurumukhi, and Roman. Normally a torn document with printed text will have similar orientation for all text present in the document. But to make our problem more challenging we intentionally prepared torn documents with multiple text orientation. Amongst all test images there were 10 document fragment images having text from multiple scripts. Those 10 document fragments were not considered during experimentation for document level script identification.

B. Accuracy at Connected/Character-component level

At the character-component stage we calculated our accuracy in two different experimental setup for both types of features. (a) A five-fold cross-validation on the character-components of all scripts found in entire training dataset. (b) First training using entire training dataset and then classifying each character-component found in all test torn document images. Below is the graph, where we depict accuracy of our scheme when applied on test dataset for both feature types. It can be noted that the Gradient-based feature slightly outperformed the Zernike moment-based feature for every class. On our test dataset, the average accuracy at character-component/connected-component level with Zernike moment-based features were 71.03% while with Gradient-based features it was 81.13%. On our training dataset at character-component/connected-component level, a 5-fold cross validation gave an accuracy of 71.33% with Zernike moment-based features and 81.39% with gradient-based features.

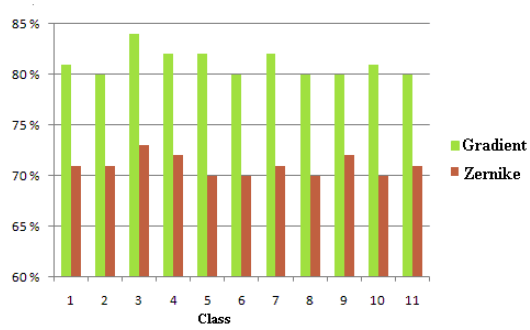


Figure 4. Percentage of accuracy at character-component/connected component level for both types of features.

C. Accuracy at word level

At word level the accuracy is calculated for all test images as follows: (a) Feature extraction is performed on each character-component found in a word.(b) Classification of each character-component is done.(c) Based on majority voting amongst all classified character-component the script type of the word is decided. (d) In case of a tie, we sum the respective confidence score of all recognized script types separately. The word is classified to the script type with maximum confidence score sum. We got an average accuracy of 94.65% with gradient features and 85.30% with Zernike features at word level. By average

accuracy we mean to say the cumulative percentage accuracy of all scripts divided by 11 (the number of scripts used in our experiment).

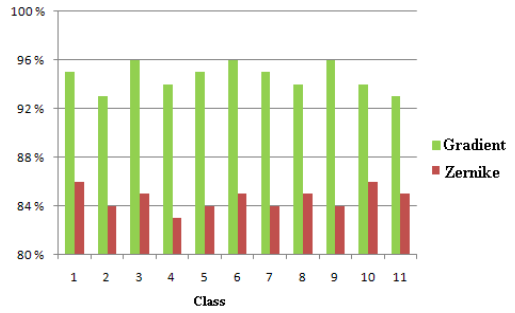


Figure 5. Accuracy at word level for both types of features.

D. *Accuracy at document level*

To calculate script identification accuracy at document level we only considered our test documents consisting of single script. At first the script of each word present in a test image are identified. Then based on majority voting amongst script type of words we conclude the script type for the document. In case of a tie, we consider it as a rejection. We obtained an accuracy of 96.7% and 98.33% at document level for Zernike moment and gradient features respectively with 0% rejection.

E. *Confidence score distribution*

Here we illustrate the distribution of confidence score of top-choice returned by our classifier, for both feature types. By confidence score, we mean to say the probability estimation of the recognized class [9]. The scores are taken during classification. We noticed that majority of correct classification with Zernike features gave a confidence score in the range of 0.6-0.69 while with gradient features it is in the range of 0.8-0.89.

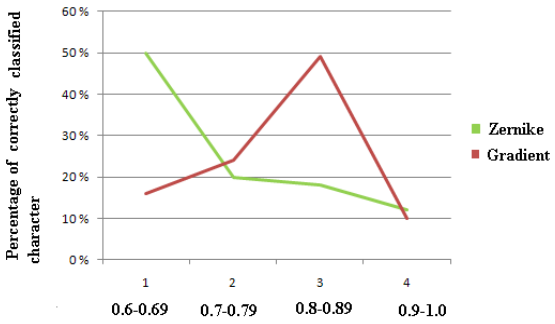


Figure 6. Distribution of confidence score in correct classification for both feature types.

VII. ERROR ANALYSIS

We analyzed the errors for both feature types. We noticed that for Zernike moment-based features, noisy images gave poor results. With gradient features, errors came mostly due to wrong orientation detection of text. This happened mostly with words when number of character-component <=2 and also with text found in the edge of torn documents. We noticed that most miss-classification occurred between Gujarati and Devnagari scripts. The reason is Gujarati characters looks very similar to a Devnagari character without headline on top.

VIII. CONCLUSION

In this article we proposed a scheme to identify 11 different scripts present in torn documents. We encountered adversaries like arbitrary orientation of text, scarcity of text in torn documents and got encouraging results using rotation dependent and independent features.

REFERENCES

[1] U. Pal, "Automatic Script Identification: A Survey", Vivek , vol. 16, 2006, pp.26-35.

[2] A. L. Spitz, "Determination of the script and language content of document images", IEEE Trans. on PAMI, vol. 19, 1997, pp. 235-245.

[3] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification", IEEE Trans. PAMI, vol. 20, 1998, pp.751-756.

[4] U. Pal, S. Sinha and B. B. Chaudhuri, "Multi-Script Line identification from Indian Documents", Proc. 7th ICDAR , 2003, pp.880-884.

[5] D. Dhanya, A. G. Ramakrishna and P. B. Pati, "Script identification in printed bilingual documents", Sadhana, vol.27, part-1, 2002, pp.73-82.

[6] L. Zhou, Y. Lu and C. L. Tan, "Bangla/English script identification based on analysis of connected component profiles", Proc. 7th DAS, 2006, pp.243-254.

[7] J. Hochberg, P Kelly, T Thomas and L. Kerns, "Automatic script identification from document images using cluster-based templates", IEEE Trans. on PAMI, vol. 19, 1997, pp. 176-181.

[8] S. B. Patil and N. V. Subareddy, "Neural network based system for script identification in Indian scripts", Sadhana, vol.27, part-1, 2002, pp.83-97.

[9] T. F. Wu, C. J. Lin, and R. C. Weng, "Probability Estimates for Multi-class Classification by Pairwise Coupling", Journal of Machine Learning Research, vol. 5, 2004, pp. 975-1005.

[10] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts", Proc. 9th ICDAR, 2007, pp. 749-753.

[11] C. Burges, "A Tutorial on support Vector machines for pattern recognition" Data mining and knowledge discovery, vol. 2, 1998, pp. 1-43.

[12] V. Vapnik, "The Nature of Statistical Learning Theory", Springer Verlag, 1995.

[13] S.Sinha, U. Pal and B. B. Chaudhuri, "Word-wise Identification from Indian documents", Lecture Notes on Computer Science (LNCS-3136), Eds. S. Marinai and A. Dengel, 2004,pp.310-321.

[14] S. Jaeger, H. Ma, and D. Doermann, "Identifying script on word-level with informational confidence", Proc. 8th ICDAR, 2005, pp.416-420.

[15] K. Franke, and J. Ruiz-del-Solar, "Soft Biometrics: Soft Computing technologies for biometric applications", in Advances in Soft Computing - AFSS 2002, LNAI 2275, Pal, N., Sugeno, M. (eds.), Springer Verlag, 2002, pp.171-177.

[16] Y.S. Lee, H.S. Koo, and C.S. Jeong, "A straight line detection using principal component analysis", Pattern Recognition Letters, 27, 2006, pp. 1744-1754.

[17] Sukalpa Chanda, Katrin Franke, and Umapada Pal, "Structural handwritten and machine print classification for sparse content and arbitrary oriented document fragments", ACM SAC 2010, 2010, pp.18-22.

[18] A. Khotanzad, and Y.H. Hong, "Invariant image recognition by Zernike moments", IEEE Transactions on PAMI, vol. 12(5),1990, pp.489–497.