Dimensionality Reduction and Feature Selection Methods for Script Identification on Document Images

Bruce Poon
School of Electrical & Information Engineering
University of Sydney
Sydney, NSW 2006, Australia
bruce.poon@sydney.edu.au

Saami Rahman, M. Ashraful Amin
Computer Vision & Cybernetics Research, SECS
Independent University
Bashundhara, Dhaka 1229, Bangladesh
saamirahman@gmail.com, aminmdashraful@ieee.org

Hong Yan
Department of Electronic Engineering
City University of Hong Kong
Hong Kong SAR, China
h.yan@cityu.edu.hk

***Abstract*—The goal of this research is to explore effects of dimensionality reduction and feature selection on the problem of script identification from images of printed documents. The k-adjacent segment is ideal for this use due to its ability to capture visual patterns. We have used principle component analysis to reduce the size of our feature matrix to a handier size that can be trained easily, and experimented by including varying combinations of dimensions of the super feature set. A modular approach in neural network was used to classify 7 languages – Arabic, Chinese, English, Japanese, Tamil, Thai and Korean.**

Keywords—feature reduction; feature selection; neural networks; principle component analysis; script identification

I. INTRODUCTION

The application of a language identification system is diverse. Almost all OCR, document indexing and classifier applications such as Google books, and all translation systems require a priori knowledge of the language used in the document. Printed documents offer a structured layout and intra-class repeatability in pattern, but they also pose the problem of repeated patterns in different languages which makes it difficult to identify discriminating patterns for inter-class classification. Much work has been done in the field of document image classification based on language. These approaches exploit different features found in distinct languages. Shwarz et al [1] used character-based features and cluster analysis to distinguish two languages – English and Arabic. Spitz [2] used topline, baseline, and different zones in a line to extract features. Such techniques require a spatial structure and fail to perform when presented with different spatial arrangements of texts. It is reported in the mentioned work that their system has performed less accurately in such conditions. Boles et al [3] used features such as bounding box of characters, text line curvature and line skew. Such approach requires extensive image processing of character segmentation.

It is observed that characters are not always well segmented, and even the best character segmentation technique poses the possibility of fragmentation of characters. The approach by Boles et al used connected component filtering to remove continuous segments of characters. This approach causes loss of information and segments and is inappropriate for use in both handwritten and printed texts as in both cases, such possibilities exist. All the features and techniques discussed above are difficult to extend to new languages. The features employed are hand-picked to work for a specific set of languages. When new languages are introduced, the same set of features may not work and a new set of features need to be derived.

Bowers et al [4] used Gabor filters in small segments of text. However, this is restricted to a defined range of text segments, and no report of applying this technique to full length document image is available. Tan [5] used template based techniques to compute the most likely script after construction of templates from clusters. These techniques are dependent on line and character segmentation. In our work, we explore the possibility of a feature that is not dependent on line or character segmentation, but can be extended to other languages to capture the inherent characteristics of a language.

II. METHODOLOGY

A. Data Set Description

For this work, we have used a standard dataset reported by Kelly et al [6]. It contains 62 images of 7 languages in total. Table 1 lists the languages and the number of images for each of them. Figure 1 shows samples from each of the seven languages.

This research is supported by the City Group Bangladesh (http://www.citygroup.com.bd).

TABLE I. LIST OF LANGUAGES IN THE DATASET

Language (7)	No. of images (62)
Arabic	10
Chinese	9
English	10
Japanese	8
Korean	8
Tamil	9
Thai	7

B. Feature Extraction Using K-Adjacent Segments

Our feature extraction process consists of two steps. The first step is edge detection [7] and the second step is to generate K-adjacent segments (K-as). This feature was first introduced by Fevrier et al [8]. We use 3-adjacent segments as 2-adjacent segments pose too much repeatability across different languages and 4-adjacent segments are too complex as we will show later in this paper. We construct a graph *G(V, E)*, where the nodes *(V)* represent each segment found from the previous step, and the edges *(E)* represent adjacency. Informally, if two segments share a common start or end point, there exists an edge between the segments represented by the nodes. We use depth first search to extract 3 consecutive segments and form 3-adjacent segments.

In order to compare different K-as, we need a numerical descriptor [8]. At the beginning, it is important to order the K-as segments $\{S_i\}$, where $i = 1..k$ in a repeatable manner so that similar K-as have the same order. In the first segment, we select those with midpoint closest to the centroid of all midpoints $\{m_i = (x_i, y_i)\}_{i=1..k}$. In the descriptor below, this centermost segment is the natural choice as a reference point for measuring the relative location of the other segments. The remaining segments take up positions 2 through k and are ordered from left to right in according to their midpoint. Note that this order is stable as no two segments can have similar location in both x and y.

Once the order is established, a K-as is a list $P = (S_1, S_2,...,S_k)$ of segments. Let $r_i = (r_i^x, r_i^y)$ be the vector going from the midpoint of s_1 to s_i. Furthermore, let θ_i and $l_i = \|s_i\|$ be the orientation from the K-as center and the length of s_i respectively.

Fig. 1. Samples from the seven languages (from left to right, then top to bottom): Arabic, Chinese, English, Japanese, Korean, Tamil, and Thai.

The descriptor of P is composed of 4k-2 values for k>1,

$$\left(\frac{r_2^x}{N_d}, \frac{r_2^y}{N_d}, ..., \frac{r_k^x}{N_d}, \frac{r_k^y}{N_d}, \theta_1, ..., \theta_k, \frac{l_1}{N_d}, ..., \frac{l_k}{N_d} \right), \quad (1)$$

where the distance N_d between the two farthest midpoints is used as normalization factor which makes the descriptor scale-invariant (Hence, both the K-as features and their descriptors are scale-invariant).

In this part, we present a measure $D(a,b)$ of the dissimilarity between two K-as P^a and P^b of the same complexity k, originally proposed by Fervier et la [8] as follows:

$$D(a,b) = w_r \sum_{i=2}^{k} \left\| r_i^a - r_i^b \right\| + w_\theta \sum_{i=1}^{k} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{k} \left| \log\left(\frac{l_i^a}{l_i^b}\right) \right| \quad (2)$$

where the first term is the difference in the relative locations of the segments. $D_\theta \in [0,\pi]$ measures the difference between segment orientations, and the last summation accounts for the difference in lengths. As segment lengths are often inaccurate, we give higher weight to the two other terms of the comparison measure: in all our experiments $w_r = 4$, $w_\theta = 2$.

To illustrate the steps in our methodology in finer details, we will use a sample image of a character and generate 3-adjacent segments from it. For better understanding, we will use the image of a character-like object rather than a character. We start the process of line fitting. Canny edge detection is used to detect roughly straight line segments and break the image structure into a combination of lines. This is demonstrated in the image shown in Figure 2 below. We now detect 3 adjacent line segments. The image contains 5 straight lines. From these 5 lines, we can obtain the triplets as such: {1, 2, 3}, {1, 3, 4}, {3, 4, 5}, {2, 3, 5}, {2, 3, 4} and {1, 3, 5}.

C. Feature Selection

After extracting contour segments, each K-as segment is represented in a 10-dimensional vector. In our set of languages, it is natural for different languages to share the same set of K-as features. We use principle component analysis (PCA) [9, 10], a statistical tool, to extract discriminating features and use them in our classifier. PCA allows us to project high dimensional data to a lower dimension. The principle

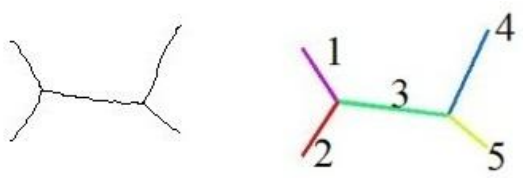

Fig. 2. A sample character and line fitting.

dimension is the axis on which the largest number of data points can be discriminated. As subsequent orders of dimension are included, more of the data points can be discriminated. In our experiment, we have found the first 3 principal components to be the most effective. We will discuss this in further detail in the results section.

We have used 300 training samples for each language. For 7 languages, we have a feature matrix of 2100 by 10. By applying PCA, we reduce this matrix to 2100 by 3.

D. Classification

Artificial neural networks are used to classify languages. From our previous research work [11], we have observed that probabilistic neural networks train faster and fit a better model compared to other popular neural network models. Moreover, it is experimentally reported that probabilistic neural networks are well known to converge [12] on large training sets.

For our simulation, we have designed our architecture with 7 probabilistic neural networks. Given a single set of network parameters such as weight and bias, it is more time consuming and difficult to draw decision boundaries that separate 7 different classes. We have used the modular neural network approach [13] motivated by the flexibility it offers to an individual network. Each of the networks only needs to classify two classes – a positive class and a negative class. Each neural network learns a single language. Given an input, all 7 networks respond with a confidence value. The trained language of the network that responds with the highest value is considered to be the language of the input image.

Features from a total of 49 images and 300 segments from each image are collected. This results in a feature matrix of size 300 by 10. To represent a language in a vector of 1 by 3000 and to train it in a neural network with other samples of the other languages is an arduous task. We have tackled this by arguing that several segments occur repetitively across different languages and we can reduce the redundancy in the features by projecting them onto a different space such that the most discriminating dimensions remain. We achieve this by using PCA.

After applying PCA, the feature matrix for one image has been reduced to 300 by 3. This gives us a handier feature size of 1 by 900. We collect more of these feature vectors and train our networks for each of the languages. Ten feature samples from each language have been trained with a total of 70 feature vectors. Concretely, our set of feature training matrix is 70 by 900.

III. Result and Analysis

At the beginning of this report, we argue that the 3-as features which have been generated capture the inherent pattern in a language. To prove this, we run experiment using different sizes of samples for the training phase.

The above plot shows the trend in accuracy with varying number of samples used for training. The accuracy begins to level off after 250 samples for each language. The best accuracy is obtained at 400 samples for each language.

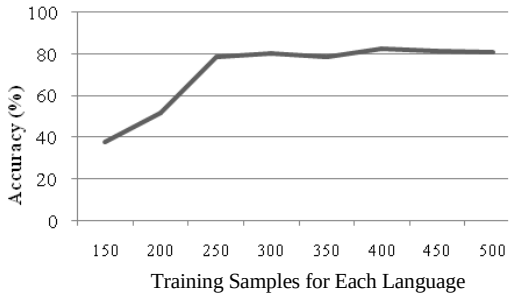

Fig. 3. Accuracy plot against number of training samples extracted from each language.

However, generating 400 samples requires exponentially longer time than extracting 300 features, hence we settle with 300 features for an optimum tradeoff between accuracy and time. Figure 4 shows the time required for feature generation against number of features generated. This graph demonstrates the trend in time required to extract 3-adj segments from a document image. This trend is expected if we observe that the 3-adj segments are generated from a graph whose size depends on the number of segments found in the document. We have experimentally observed that for 300 3-adjacent segments, a graph of 3500 by 3500 should be generated. This is a large graph and running repeated depth-first search to extract 3-adjacent segments on this is computationally intensive. This task is increased in many folds for a larger number of 3-adjacent segments such as 400 which requires a much larger graph to be traversed.

A. Number of Principal Components

When projecting using PCA, we have to select the number of dimension to project on. The appropriate number of choice is dependent on the data being projected, and can be determined experimentally. We have conducted several trials of our experiment with varying number of principle components. The corresponding accuracies are demonstrated in the plot below.

Figure 5 shows that the classifier has consistent learning performance, starting from the first 3 principle components to the 6^{th} principle component being used. We deduct that the 4^{th}, 5^{th} and 6^{th} component do not add any significant discriminating feature information to the classifier and hence no drastic changes in accuracy are observed. Adding the 4^{th} component results in a very slight improvement in accuracy, but we settle

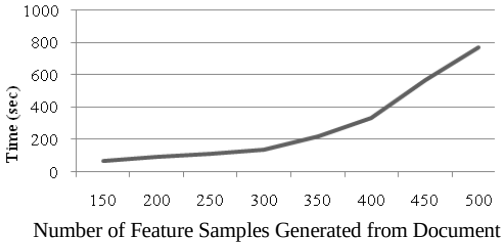

Fig. 4. Plot for time required versus number of feature samples generated from a document image.

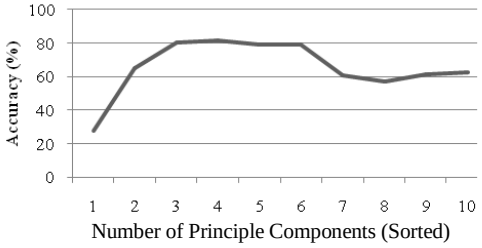

Fig. 5. Changes in Accuracy on Varying Number of Principle Components.

with the 3rd component in lieu of reduced computation time and cost. When the 7th component is added, there is a drop in accuracy to 60.8%. Further addition of components leads to a consistent accuracy around 60%. We predict that adding further dimensions to our feature causes the classifier to be fed with misleading data and creates confusion for the classifier. The projection on the 7th dimension and further contain an overlap of the classes which makes it difficult for the classifier to perform well. To confirm our prediction, we choose different numbers of principle components for classifying them.

Figure 6 shows the obtained accuracies with different choices of principle components. It is established that components 1, 2 and 3 are the strongest in their discriminating power, whose performance measure is added to the plot for reference. We observe the accuracy by adding the 7th, 8 and 9th component to see the effect on the discriminating power of the first 3 components. The results support our prediction that the projection on 7th component results in confusion and overlapping of classes for the classifier. Adding the 8th and the 9th component to the first 3 principle components comparatively reduces the accuracy less. Using only the 7th, 8th and 9th component results in a low accuracy of 47.9%. Since the 8th and 9th component are weaker than the first 3 components, the effect of adding the 7th component is more pronounced.

B. The Confusion Matrix

The confusion matrix of various classifiers is shown in Table 2 .According to this table, the largest confusion arises when classifying a Tamil document from a Thai document. An inspection of Thai and Tamil language data shows that both these languages are very similar visually. In some cases, it is difficult for humans to classify Tamil and Thai documents based on visual inspection. Since our feature captures visual information rather than character information, this confusion seems unavoidable.

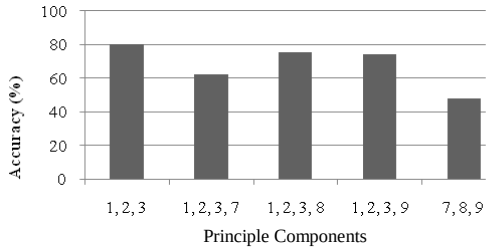

Fig. 6. Accuracy Plot Against Different Choices of Principle Components.

TABLE II. CONFUSION MATRIX FOR VARIOUS CLASSIFIERS. LEGENDS: A (ARABIC), C (CHINESE), E (ENGLISH), J (JAPANESE), K (OREAN), TM (TAMIL), TH (THAI).

	A	C	E	J	K	TM	TH
A	93.8	1.3	0.6	1.9	0.4	1.1	0.9
C	1.3	77	1.1	10.6	6.4	1.7	1.9
E	0.6	1.1	91.9	1.6	0.5	1.9	2.4
J	0.5	10.6	0.4	78.6	7.1	1.5	1.3
K	0.4	6.4	0.3	7.1	81.2	3.4	1.2
TM	1.1	1.7	1.9	1.5	3.4	69.6	20.8
TH	0.9	1.9	2.4	1.3	1.2	20.8	71.5

The second most confusing classification is caused when classifying Chinese from Japanese documents if the Japanese documents consist mainly of Kanji character sets. This can be explained by the fact that Japan borrowed Chinese characters to form her own Kanji character sets which are similar to or the same as the Chinese characters.

The best classification is for Arabic at 93.8%. Arabic is visually very unique from all other languages. This avoids confusion with other documents and results in a much better classification when compared to other languages.

From the confusion matrix, the mean diagonal is 80.5% which we report as our average accuracy.

Figure 7 shows the 10-fold validation for our system. It shows that the performance of our system is consistent. The training accuracy is higher than the testing accuracy. We attribute this to the nature of probabilistic neural networks, whose design allows them to fit a model much better during the training phase than compared with the testing phase. We intend to explore this variation in our future work.

IV. CONCLUSION

We have proposed the use of 3-adjacent segments for language classification with the use of PCA for feature reduction and a modular neural network design for classification. The results show a consistent performance at an average accuracy of 80.5%. The strength of our system is that it

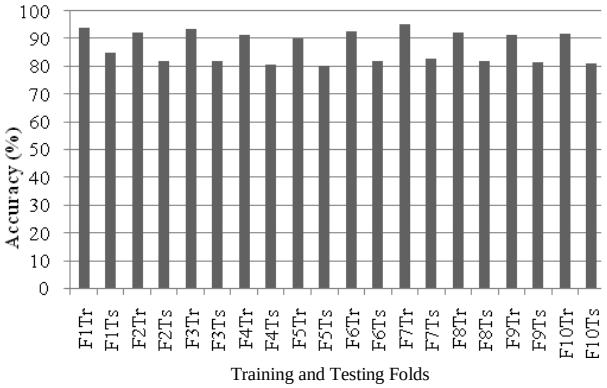

Fig. 7. 10-Fold Validation of the System.

uses a feature set which captures the visual arrangement in a language's character set, rather than the character information. It is easy to extend the system for further languages. The accuracy found is for languages that share a great deal of visual similarity. For languages which are dissimilar to each other to a great extent, we have found accuracy is greater than 90%. There is no report of any time measure in this area of research. We have reported the times of our system's training phase for a varying number of samples and believe that this is smaller than those in other reported works since our feature extraction phase samples a defined number of features rather than processing the whole document.

This research work can be further extended and improved. For the samples of the seven languages we use, the sentences are being read from left to right. In Chinese and Japanese languages, modern way is being read from left to right. However, they can also be read from top to bottom as well as from right to left. More works need to be done to ensure the methods we use can identify those languages which can also be read in a non-common way as described above.

This work can also be extended to consider time measure for a defined number of features as well as for processing the whole documents.

ACKNOWLEDGMENT

The major work in this paper had been presented in International Conference on Information Technology and Application (ICITA 2013), 1 – 4 July, Sydney, Australia. In addition, we addressed those issues arose during ICITA 2013 for further improvement in our work.

REFERENCES

[1] R. Shwarz, J. Makhoul, and I. Bazzi, "An Omnifont Open-Vocabulary OCR System for English and Arabic," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 6, pp. 495–504, 1999.

[2] A. L. Spitz, "Determination of the Script and Language Content of Document Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 3, pp. 235245, 1997.

[3] W. Boles, S. Sridharan, and A. Busch, "Texture for Script Identification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 17201733, 2005.

[4] K. Bowers, M. Cannon, P. Kelly, and J. Hochberg, "Script and Language Identification for Handwritten Document Images," Los Alamos National Laboratory, Los Alamos, 1997.

[5] T. Tan, "Rotation Invariant Texture Features and Their Use in Automatic Script Identification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 7, pp. 751756, 1998.

[6] P. Kelly, T. Thomas, L. Kerns, and J. Hochberg, "Automatic Script Identification from Document Images Using Cluster-Based Templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 176181, 1997.

[7] J. Canny, "A Computational Approach To Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679698, 1986.

[8] L. Fevrier, F. Jurie, C. Schmid, and V. Ferrari, "Groups of adjacent contour segments for object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 1, pp. 3661, 2008.

[9] L. I. Smith, "A Tutorial on Principal Components Analysis," Cornell University, 2002. [Online]. Available:

http://www.sccg.sk/~haladova/principal_components.pdf

[10] L. J. Williams and H. Abdi., "Principal Component Analysis," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, pp. 433459, 2010.

[11] S. F. Miskhat, M. Ridwan, E. Chowdhury, S. Rahman, M. A. Amin, "Profound Impact of Artificial Neural Networks and Gaussian SVM Kernel on Distinctive Feature Set for Offline Signature Verification," in Proceedings of the International Conference on Infomatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, pp. 940945, 2012.

[12] K. Cannon and V. Cheung, University of Toronto, Mar 2012. [Online]. Available:

http://www.psi.toronto.edu/~vincent/research/presentations/PNN.pdf

[13] H. Irani, H. R. Pourreza, and O. Mirzaei. "Offline Signature Recognition using Modular Neural Networks with Fuzzy Response Integration," in Proceedings of the 2011 International Conference on Network and Electronics Engineering, vol. 11, Kuala Lumpur, Malaysia, pp. 5359, 2011.

