On-line Script Recognition *

Anoop M. Namboodiri and Anil K. Jain
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{anoop, jain}@cse.msu.edu

Abstract

Automatic identification of handwritten script facilitates many important applications such as automatic transcription of multi-lingual documents and search for documents on the Internet containing a particular script. The increase in usage of handheld devices which accept handwritten input is creating a huge volume of handwritten data. We propose a method to classify words and lines in an on-line handwritten document into Arabic, Cyrillic, Devnagari, Han, Hebrew and Roman scripts. The proposed classification system, based on spatial and temporal features of the strokes, attained an overall classification accuracy of 86.5% *at the word level on a dataset containing* $13,379$ *words. The classification accuracy improves to* 95% *as the number of words in the test sample is increased to five and to* 95.1% *for complete text lines.*

1. Introduction

With the increase in popularity of portable computing devices such as PDAs and handheld computers [3], non-keyboard based methods for data entry are receiving more attention. The most promising options are pen-based and speech-based inputs. Pen-based input devices generate handwritten documents which have on-line or dynamic (temporal) information encoded in them. As computing platforms which use pen-based input such as the IBM Thinkpad TransNote [2] and Tablet PCs [4], which generate on-line documents, become available and affordable, the demand for algorithms which can process on-line data for efficient storage and retrieval also increases.

On-line documents may be written in different languages and scripts. Figure 1 shows an example of a document page containing six different scripts. A script is defined as a graphic form of a writing system [5] such as the *alphabetic* system, which is adopted by scripts like Roman and Greek, or the *syllabic-alphabetic* system, which is adopted by most Indian scripts. A specific script like Roman may be used by multiple languages such as English, German and French. During the evolution of languages, existing scripts were adopted or modified by many languages to suit their specific words and sounds [9]. Due to such interactions, the scripts of many languages are either identical or have only minor variations.

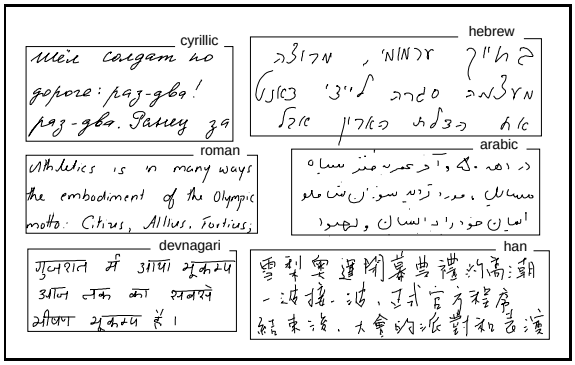


Figure 1. A multi-script on-line document.

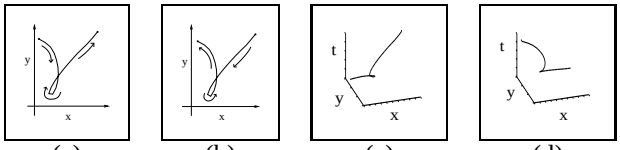
Nakanishi [11] gives a comprehensive survey of the scripts currently used in the world. The six scripts, Arabic, Cyrillic, Devnagari, Han, Hebrew and Roman cover the languages used by a majority of the world population. Most of the other scripts are used exclusively by specific languages. Based on the above observation of the relationship between languages and their scripts, the six most popular scripts, Arabic, Cyrillic, Devnagari, Han, Hebrew and Roman, were chosen for our on-line classification study (see figure 1). The general class of Han-based scripts include Chinese, Japanese and Korean. In this work, we have used only Chinese characters and hence we use the term *Han Script* to refer to the Chinese character set. Devnagari script is used by many Indian languages, including Hindi, Sanskrit and Marathi. Arabic script is used by Arabic, Farsi, Urdu,

*Supported by IBM University Partnership Program

etc. Roman script is used by many European languages like English, German, French and Italian.

Most of the published work on automatic script recognition deals with off-line documents, i.e., documents which are either handwritten or printed on a paper and then scanned to obtain a two-dimensional digital representation. Approaches for script identification in printed documents can be found in Spitz [14], Jain et al. [8], Tan [15] and Pal and Chaudhuri [13]. For off-line handwritten script identification one may refer to Hochberg et al. [6]. Note that some of the previous work on on-line documents (e.g., [12]) uses the term *on-line* to refer to documents on the Internet where the goal is to infer the language of a character-coded text document. The only work in processing multi-lingual on-line documents that we are aware of is by Lee et al. [10], which attempts to do recognition of multiple languages simultaneously using a hierarchical Hidden Markov Model.

The most important characteristic of on-line documents is that they capture the temporal sequence of strokes[1]. This allows us to analyze the individual strokes and use the additional temporal information for both script identification as well as text recognition. Unfortunately, the temporal information also introduces additional variability to the handwritten characters which creates large intra-class variations of strokes in each of the script classes. Figure 2 shows two samples of the character '**r**' (different writing directions), with and without the temporal information. Even though the spatial representations in 2(a) and 2(b) look similar, the temporal differences introduce large intra-class variability in the on-line script (see 2(c) and 2(d)).



(a) (b) (c) (d)

Figure 2. On-line Script Variability. Off-line representations of the letter 'r' in (a) and (b) look similar, but the corresponding on-line strokes in (c) and (d) look very different. Writing directions in (a) and (b) are different.

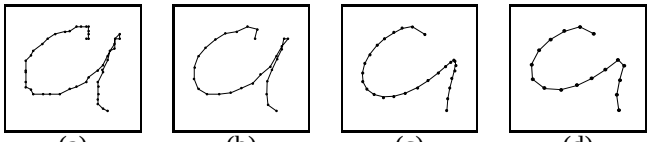
We attempt to solve the problem of script recognition, where either an entire document or a part of it (upto word level) is classified into one of the six scripts mentioned above. Jain et al. [7] discusses extraction of text regions from an on-line document. The problem of identifying the actual language often involves recognizing the text and identifying specific words or sequence of characters, which is beyond the scope of this paper.

[1] A stroke is defined as the locus of tip of the pen from pen-down to the next pen-up position.

2. Data Collection and Pre-processing

The data used in this paper was collected using the *CrossPad* ® [1]. The *CrossPad* has a pen and paper interface along with the ability to digitally capture the (x, y) position of the pen tip at a resolution of $254\ dpi$. The pen position is sampled at a constant rate of $132\ Hz$. We must point out that the actual device for data collection is not important as long as it can generate a temporal sequence of x and y positions of the pen tip. The users were asked to write one page of text on a ruled paper in a particular script, with each page containing approximately 20 lines of text. No restriction was imposed on the content or style of writing.

During pre-processing, the individual strokes are resampled to make the sampled points equidistant. The strokes are then smoothened using a Gaussian (lowpass) filter to reduce noise in sampling. The strokes are again resampled to make the points equidistant. During the lowpass filtering and resampling operations, the *critical points* in a stroke are retained. A critical point is defined as a point (x, y) in the stroke where the sign of Δx or Δy changes, in addition to the pen-up and pen-down points. Figure 2 shows an example of the preprocessing operations.



(a) (b) (c) (d)

Figure 3. Pre-processing. (a) original stroke; The dots represent the sampling points. (b) equidistant resampling. (c) lowpass filtering. (d) second resampling.

The data available to a script recognizer is usually a complete handwritten page or a subset of it. To identify the individual lines, first the inter-line distance is estimated. The inter-line distance, d, is defined as the distance between successive peaks in the autocorrelation of the y-axis projection of the text. Lines are identified by finding valleys in the projection, keeping the inter-line distance as a guiding factor. The temporal information from stroke order is used to disambiguate strokes which fall across line boundaries and also to correctly group small strokes which may fall into an adjacent line. The segmentation of a line into words is done using an x-axis projection of the text in the line. The gaps in the projection are noted as word boundaries and the strokes which fall between two boundaries is collected and labeled as a word. Figure 4 shows the output of our line and word detection algorithms for the multicolumn document of figure 1, where the text lines are underlined and alternate words are shown in dark and light shades.

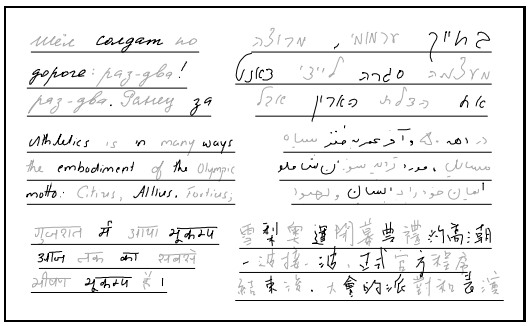


Figure 4. Identifying *text lines* and *words*.

3. Feature Extraction

It is helpful to study the general properties of each of the six scripts for feature extraction. (i) Arabic lines are written from right to left. A typical Arabic character contains a relatively long main stroke, drawn from right to left, along with one to three dots. The length of the strokes vary considerably. (ii) Cyrillic script, although similar to the cursive Roman script, has its individual characters connected together to form a long stroke in a word. The absence of delayed strokes (e.g., the horizontal stroke of the letter *t* drawn after writing the whole word) is another notable difference. (iii) The most important characteristic of Devnagari script is the horizontal line present at the top of each word, called '*Shirorekha*', usually drawn after the whole word is written. (iv) Characters of Han script are composed of multiple short strokes, usually drawn from top to bottom and left to right within a character. The direction of writing of words in a line is either left to right or top to bottom. The database used in this study contains Han script text with horizontal lines only. (v) Words in a line of Hebrew script are written from right to left as in Arabic script. The most distinguishing factor of Hebrew from Arabic is that the strokes are more uniform in length. (vi) The length of the strokes in Roman script tend to fall between that of Devnagari and Cyrillic scripts. The features were extracted either from the individual strokes or from a collection of strokes. Here, we describe each of the feature and how it is computed.

1. *Horizontal Inter-stroke Direction (HID)*: This captures the "direction" of writing within a line.

$$HID = \sum_{i=1}^{n-r} dir(i, i+r),$$

where $dir(i, j)$ is $+1$, if the x coordinate of the pen-down position of the stroke i is less than that of stroke j, and -1 otherwise, n is the number of strokes in the pattern and r is set to 3 to reduce errors due to abrupt changes in direction between successive strokes.

2. *Average Stroke Length (ASL)*: The ASL is defined as the average length of the individual strokes (no. of points in a stroke) in the pattern.

3. *Shirorekha Strength*: *Shirorekha Strength* is defined as the ratio of the sum of the bins corresponding to line orientations between $-10°$ and $10°$ to the sum of all the bins in the Hough transform space, $H(r, \theta)$.

4. *Shirorekha Confidence (SC)*: Shirorekhas span the width of a word, occur at the top of the word and are horizontal. Hence the confidence (C) of a single stroke, s, is defined as:

$$C(s) = \frac{W(s)}{W(pattern)} * \frac{\bar{Y}(s)}{H(pattern)} * \left(1 - \frac{H(s)}{W(s)}\right),$$

where $W(s)$ is the width of s (length along x-axis), $H(s)$ is the height of s (length along the y-axis), and $\bar{Y}(s)$ is the average of the y-coordinates of the stroke points. $C(s)$ is set to zero, if its computed value is negative. For an n-stroke pattern, the *SC* is computed as the maximum value for C among all its component strokes.

5. *Stroke Density*: This is the number of strokes per unit length (along x-axis) of the pattern.

6. *Aspect Ratio*: Aspect Ratio is the ratio of the width to the height of a pattern.

7. *Reverse Distance*: This is the distance by which the pen moves in the direction opposite to the normal writing direction. Note that the normal writing direction is different for different scripts.

8. *Average Horizontal Stroke Direction (AHSD)*: Horizontal Stroke Direction (HD) of a stroke, s, is defined to be $+1$ if the x coordinate of the pen-down position of s is less than that of its pen-up position, and -1, otherwise. For an n-stroke pattern, AHSD is computed as the average of the HD values of its component strokes.

9. *Average Vertical Stroke Direction (AVSD)*: It is defined similar to the AHSD. For AVSD, we consider the y coordinates of the pen-down and pen-up points, instead of x coordinates.

4. Experimental Results

The data set consisted of $13,379$ words; $1,423$ Arabic, $1,002$ Cyrillic, $3,173$ Devnagari, $1,981$ Han, $2,261$ Hebrew and $3,575$ Roman. The data was randomly divided into 5 groups and a 5-fold cross validation (repeating the experiment 5 times, each time using a different group as test set and the rest of the data for training) was done. The

Classifier	Remarks	Error Rate
1-NN	No Normalization	35.8 %
1-NN	Normalized Features	17.6 %
5-NN	Normalized Features	15.4 %
Bayes Quadratic	Gaussian with Full Covariance	22.9 %
Mix. of Gaussian	Diagonal Covariance	25.5 %
Decision Tree	C5.0	16.1 %
Neural Network	1 hidden layer with 25 Nodes	14.3 %
SVM	RBF Kernel	13.5 %

Table 1. Word-level classification.

error rates reported are the averages of the five trials. Table 1 shows the performance of different classifiers. The normalization mentioned in table 1 ensures that each feature has zero mean and unit variance. The SVM-based classifier gives the best performance of 86.5% for word-level classification. The classification accuracy increases as the number of words in a sample increases (evidence accumulation); with 5 words in a test sample, we obtain an accuracy of 95%. Figure 5 gives an example of the output of the script classifier on a test page containing all the six scripts. Note that a few Roman words written using short strokes are misclassified as Han and some Roman words written in a cursive manner are misclassified as Cyrillic.

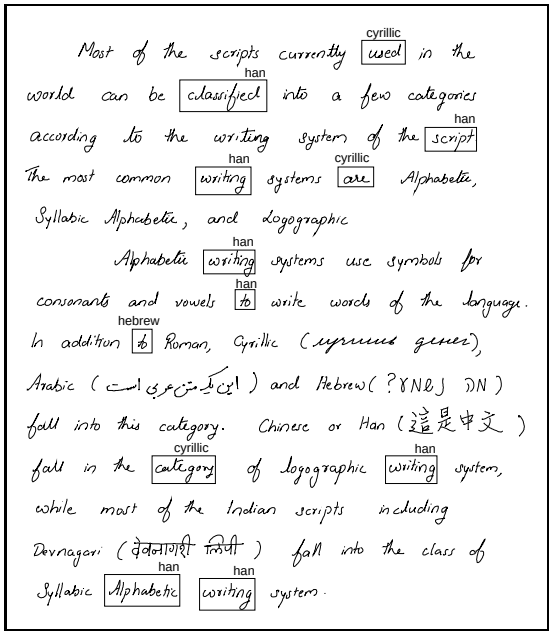


Figure 5. Example of script classification. Misclassifications are boxed and labeled.

5. Conclusions and Future Work

The papers presents a script identification algorithm to recognize 6 different scripts in an on-line document The classification accuracy is 86.5% at word level and it increases to 95% as the number of words in a sample in increased to 5 and 95.1% for complete lines (7 words on an average). Techniques to combine the results of multiple classifiers are being investigated.

References

[1] IBM Pen Technologies. http://www.research.ibm.com/handwriting/.

[2] IBM ThinkPad TransNote. http://www.pc.ibm.com/us/thinkpad/transnote/.

[3] Pen Computing Magazine: PenWindows. http://www. pencomputing.com/PenWindows/index.html.

[4] Windows XP Tablet PC Edition Homepage. http://www. microsoft.com/windowsxp/tabletpc/default.asp.

[5] F. Coulmas. *The Blackwell Encyclopedia of Writing Systems*. Blackwell Publishers, Malden, Massachusetts, 1999.

[6] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly. Script and language identification for handwritten document images. *International Journal on Document Analysis and Recognition*, 2(2):45–52, February 1999.

[7] A. K. Jain, A. M. Namboodiri, and J. Subrahmonia. Structure in on-line documents. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 844–848, Seattle, Washington, September 2001.

[8] A. K. Jain and Y. Zhong. Page segmentation using texture analysis. *Pattern Recognition*, 29(5):743–770, May 1996.

[9] H. Jensen. *Sign, Symbol and Script: An Account of Man's Effort to Write*. George Allen and Unwin, London, 3^{rd} edition, 1970.

[10] J. J. Lee and J. H. Kim. A unified network-based approach for online recognition of multi-lingual cursive handwritings. In *Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 393–397, Colchester, England, September 1996.

[11] A. Nakanishi. *Writing Systems of the World*. Charles E. Tuttle Company, Tokyo, 1999.

[12] N. Nobile, S. Bergler, C. Y. Suen, and S. Khoury. Language identification of on-line documents using word shapes. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 258–262, Ulm, Germany, August 1997.

[13] U. Pal and B. B. Chaudhuri. Script line separation from Indian multi-script documents. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, Bangalore, India, September 1999.

[14] A. L. Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):235–245, March 1997.

[15] T. N. Tan. Rotation invariant texture features and their use in automatic script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):751–756, July 1998.