

## Identification of Japanese and English Script from a Single Document Page

S. Chanda, U. Pal and F. Kimura\*

Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-108, India

\*Graduate School of Engineering, Mie University, 1577 Kurima-machiya, Tsu, 514-8507, Japan

E-mail: [kimura@hi.info.mie-u.ac.jp](mailto:kimura@hi.info.mie-u.ac.jp)

### Abstract

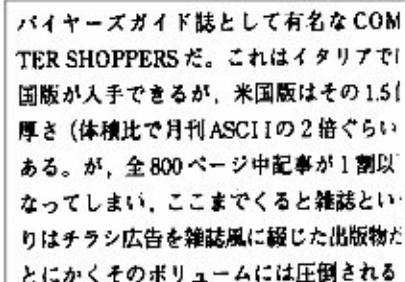
*In Japanese documents, a single text line of a page may contain both Japanese and English scripts. For the Optical Character Recognition of such a document page it is better to identify Japanese and English script portions at first, and then to use individual OCRs of these two scripts on their respective identified portions to get higher OCR accuracy. In this paper, an automatic technique for identification of Japanese and English script portions from a single line of a printed document page is proposed. To the best of our knowledge this is the first work of its kind. Here, at first, the document is segmented into lines and then lines are segmented into characters. In the proposed scheme, individual scripts are identified using combination of different features obtained from structural shape of characters, pitch information, topological properties, water reservoir concept etc. Based on the experiment on 11304 characters, we obtained 98.79% identification accuracy from the proposed scheme.*

### 1. Introduction

There are many Japanese documents where a single document page contains both Japanese and English text. An example of such document is shown in Fig.1. Japanese OCR software has two reading modes, i.e. Japanese mode and English mode. The English mode is aimed to recognize characters used in English text (alphanumeric characters and symbols), while the Japanese mode is aimed to recognize all characters used in Japanese text (alphanumerals as well as symbols of Kanji, Hiragana and Katakana scripts). Since the English mode is specialized for segmentation and recognition of English characters, it performs better in English region than Japanese mode. However, the English mode is not available for Japanese-English mixed text, thus the recognition accuracy of the English region is relatively low. From the experiment we notice that performance of existing commercial Japanese OCR software reduces when the input Japanese text includes English words/sentences, computer programs and commands, etc. In this paper, an automatic technique for identification of Japanese and English scripts is proposed so that individual OCR systems can be run on their respective modes in this identified portion to get better OCR accuracy.

Among the pieces of earlier work Spitz [1] developed a method to separate Han based or Latin based script separation. He used optical density distribution of characters and frequently occurring word shape characteristics for the purpose. Using cluster based templates, an automatic script identification technique has been described by Hochberg *et al.* [2]. Ding *et al.* [3] proposed a method for separating two classes of scripts: European (comprising Roman and Cyrillic scripts) and Oriental (comprising Chinese, Japanese and Korean scripts). Using fractal-based texture features, Tan [4] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text. Pal *et al.* [5] proposed a line-wise script identification scheme from tri-language documents of Indian scripts. Pan *et al.* [8] discussed a method for script identification using steerable Gabor filters. Dhanya *et al.* [9] used Linear Support Vector Machine (LSVM), K-NN and Neural Network (NN) classifiers on Gabor-based and zoning features to classify Tamil and English scripts. Recently, Jaeger *et al.* [11] used K-NN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean and Hindi texts.

In this paper, a scheme for identification of different portions of Japanese and English scripts from a single text line is discussed. Using different features obtained from structural shape, pitch information, topological properties, water reservoir concept etc., the proposed scheme is developed.



バイヤーズガイド誌として有名なCOMPUTER SHOPPERSだ。これはイタリアで1000部が入手できるが、米版はその1.5倍の厚さ(体積比で月刊ASCIIの2倍ぐらいある。が、全800ページ中記事が1割以下はチラシ広告を雑誌風に綴じた出版物だ。とにかくそのボリュームには圧倒される

Fig.1. Example of a document containing both English and Japanese characters.

## 2. Evaluation of OCR performance on Japanese-English mixed text

Eight test sheets containing both Japanese and English characters are used in the performance evaluation. Table 1 shows the number of characters present in Japanese and English regions of the respective test sheets.

Four typical Japanese OCR softwares A, B, C, D are used in the evaluation test. Because of the non-disclosure agreement between the company and laboratory we cannot provide the names of the four softwares used here. Table 2 shows the accuracy of character recognition obtained by each of the four OCR softwares on the above test sheets. This table shows that the character recognition error rates in English region reduces 6.95% (18.10% to 11.15% on average) by the use of the English mode instead of Japanese mode. So, initially if we identify the scripts and then run OCR of individual mode, better results can be achieved and hence we propose a scheme for automatic identification of Japanese and English scripts.

**Table 1. Number of characters in each region of Japanese-English mixed text sheets**

Test sheet	English region	Japanese region	Total
Windows manual	215	528	743
Magazine (ASCII)	157	893	1050
Advertisement	263	972	1235
Magazine (Nikkei Byte)	643	1065	1708
Magazine (Interface)	253	295	548
Magazine (ASCII)	98	1133	1231
Magazine (Interface)	349	1317	1666
Magazine (Nikkei Byte)	190	796	986
Total	2466	8838	11304

**Table 2. Accuracy of character recognition for Japanese-English mixed text (%)**

OCR software	Japanese region	English region	
	Japanese mode	Japanese mode	English mode
A	92.88	78.85	90.44
B	91.23	80.20	92.51
C	95.89	89.38	84.73
D	88.41	79.19	87.73
Average	92.10	81.90	88.85

## 3. Preprocessing and Features Extraction

The digitized images are in gray tone and we have used a histogram based thresholding approach to convert them into two-tone images [1 and 0, where 1 represents foreground pixel]. The digitized image may contain spurious noise pixels and irregularities on the boundary of the characters, leading to undesired effects on the system. For removing these noise pixels we have used a simple and efficient method due to [7]. The lines are segmented from the documents by finding the valleys of the projection profile computed by counting the number of black pixels

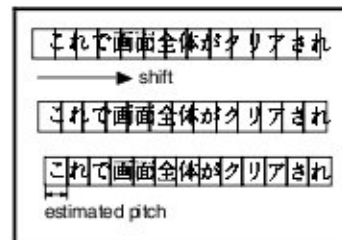
in each row. The valley between two consecutive peaks in this profile denotes the boundary between two text lines. A text line can be found between two consecutive boundary lines. After a text line is segmented, it is scanned vertically. If in one vertical scan two or less black pixels are encountered then the scan is denoted by 0. Else, the scan is denoted by the number of black pixels. In this way a vertical scanning histogram is constructed. Now, if in the histogram there exist a run of at least  $k_1$  consecutive 0's then the midpoint of that run is considered as the boundary of a character. The value of  $k_1$  is taken as statistical mode of the horizontal and vertical black runs obtained by scanning the characters of a line in row and column-wise.

We will now discuss some of the features used in our script identification.

### 3.1 Character Pitch Feature

The height and width of printed Japanese characters are correlated, and the characters are usually aligned in fixed pitch (by pitch we mean the distance between the CGs of two consecutive characters). This property can be utilized to estimate the pitch of character alignment and to detect the fixed pitch regions. Fix pitch region concept is not present in English and this pitch feature helps us to identify Japanese text from mixed Japanese-English text.

The pitch of character alignment in each line is estimated by the following two procedures. (1) Given a width of rectangular frame of a character, a ladder of horizontally aligned frames is shifted from left to right (see Fig. 2). The width of the frame ranges from 80 to 125% of the height of characters, and the horizontal displacement of the ladder ranges from 0 to 100% of the width. (2) The width of the frame that minimizes the number of black pixels on the edges of the ladder found in (1) is defined as the estimated pitch. The number of black pixels on the edges of the ladder is calculated using horizontal pixel projection of the text line.

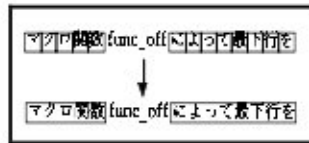


**Fig.2. Estimation of character pitch**

Shifting the ladder with estimated frame width from left to right on the text line, a region of characters enclosed in five or more successive frames without intersecting any black pixel is detected as a fixed pitch region (see Fig.3). Also, at both ends of the text line, a region of characters enclosed in three or more successive frames is detected as a fixed pitch region. In Fig.3 fix pitch regions of the first line are detected and marked by rectangle in the second line of this image. If in a text line we get fix pitch region with five or more successive frames in the middle of the text line we identify that portion as Japanese. If three or



more successive frames are achieved at both ends of a text line then that portions are also identified as Japanese.



**Fig.3. Detection of fixed pitch region. Fix pitch regions are marked by rectangle in the second line of the image.**

### 3.2 Water Reservoir Principle Based features:

The water reservoir principle is as follows. If water is poured from one side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [6]. This water reservoir concept helps us to find some distinguishable features, which are very helpful for Japanese and English script separation.

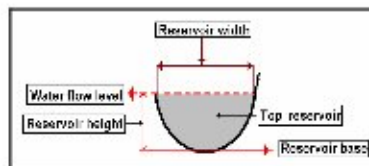
We will discuss now some properties of water reservoir.

**Top (bottom) Reservoir:** The reservoir obtained when water is poured from top (bottom) of the component is called top (bottom) reservoir. (A bottom reservoir of a component is visualized as top reservoir when water will be poured from top after rotating the component by  $180^\circ$ ).

**Left (Right) Reservoir:** The reservoir obtained by pouring water from left (right) side of the component is called top left (right) reservoir. For an illustration see Fig.4. Here top, bottom, left and right reservoirs of some English characters are shown.



**Fig.4. Top, Bottom, left and right reservoirs are shown in different English characters.**



**Fig.5. Reservoir base, flow-level, reservoir height and width, are shown on a top reservoir of a component. Here reservoir is marked by grey-shades.**

**Water flow level:** The level from which water overflows from a reservoir is called as water flow level of the reservoir (see Fig.5).

**Reservoir base-line:** A line passing through the deepest point of a reservoir and parallel to water flow level of the reservoir is called as reservoir base-line (see Fig.5).

**Height of a reservoir:** By height of a reservoir we mean the depth of water in the reservoir. In other words, height of a reservoir is the normal distance between reservoir base-line and water flow level of the reservoir (see Fig.5).

**Width of a reservoir:** By width of a reservoir we mean the maximum distance among the horizontal distances obtained between two reservoir surface points at the flow level (see Fig.5).

All reservoirs obtained from a direction of a component are not considered for future processing. The reservoirs having heights greater than a threshold  $T_1$  are only considered. This threshold value is obtained from the experiment and is considered as 20% of the respective text line height on which the component lies.

Some of the features extracted from water reservoir concept are (i) type of water reservoir obtained (top, bottom, left or right) (ii) water flow level (iii) water reservoir height (iv) reservoir width (v) structural shape of water reservoir surface, etc.

### 3.3 Component overlapping feature:

Component overlapping feature is another distinct and simple feature and it is used for the identification between English and Japanese texts. In English text, vertical overlapping of two components is absent except the characters 'i' and 'j' but in Japanese there are many characters where one component overlaps vertically with other component. For illustration see Fig.6 where a Japanese and an English line is shown. From this figure it can be seen that component overlapping frequently occurs in Japanese characters. To compute this feature, we draw the bounding box of each component and check whether this bounding box vertically overlaps or not. Although two English characters 'i' and 'j' have vertical overlapping, these two characters have distinct behaviors that separate these two characters from Japanese. For these two English characters a vertical line-like structure is present in the position just below the upper component and difference of the width of the two components (dot and main part of i) is less than stroke width ( $R_L$ ) of the character. Here stroke width  $R_L$  is the statistical mode of the black run lengths of the character. For a character,  $R_L$  is calculated as follows. The character is, at first, scanned row-wise (horizontally) and then column-wise (vertically). If  $n$  different runs of lengths  $r_1, r_2, \dots, r_n$  with frequencies  $f_1, f_2, \dots, f_n$ , respectively are obtained after these two scanning from the character, then value of  $R_L$  will be  $r_j$  if  $f_j = \max(f_i), j = 1, 2, \dots, n$ .



**Fig.6. Text lines with the bounding box of each component (a) Japanese (b) English lines.**

### 3.4 Crossing count feature:

Number of crossing obtained in vertical as well as horizontal direction is an important feature for Japanese and English script. This is because of the presence of Kanji characters in the Japanese text. Analyzing Kanji characters we note that most of the Kanji characters have more than three crossing count both in column-wise as well as row wise. Some of the Kanji characters are shown in Fig.7.



Fig.7. Some examples of Kanji characters.

For the use of crossing count feature we, at first, find maximum number of black run obtained by row-wise scanning. Also we find maximum number of black run obtained by column-wise scanning. If the number of maximum black run in both the scanning is  $\geq 3$  we mark the character as Japanese. Else, if the number of maximum black run obtained only in row-wise is  $\geq 3$  then most of the cases it will be Japanese but there are some English characters "M", "m", "N", "w", "W" etc., where this property may satisfy. To identify these English characters from Japanese we have used water reservoir concept based features. It can be noted that in each of these English characters there are two reservoirs with equal height and shape. Also water flow level coincides with character boundary for these characters. For illustration see Fig.8.



Fig.8. Two similar reservoirs in each of the characters are shown by different gray-shades. Water flow levels are shown by arrows.



Fig.9. Illustrations of profile feature detection.

### 3.5 Profile feature:

We use profile-based method for resolving confusions occur during script identification. Suppose each character is located within a rectangular boundary, a frame. The horizontal or vertical distances from any one side of the frame to the character edge are a group of parallel lines, which we call the *profile*. We compute left profile and right profile of a character and we observe the behavior of the profiles. In some cases, we compute the number of transitions (change of mode) in the profiles in a specific region of the characters and use this value in our identification scheme. This profile also helps us to detect whether vertical-line-like structure present in a character or not. If all left (or right) profiles of this region have unique behavior (either all increasing or all decreasing mode) we can say that a vertical line exists in that character. For example see Fig.9, where left and right profile of the italics character 'P' is shown. Here all the left profiles in the character are in decreasing mode from top to bottom. Hence we assume vertical line exist in the character. Note that even if a character is italics this method can also detect vertical line-like structures. Sometimes we also compute top and bottom profiles of a character for identification.

## 4. Script identification

Based on the above features, we use a binary tree classifier for the script identification. The features are chosen at a non-terminal node to get optimum tree [7]. In the tree sometimes a script belongs to both the sub-groups of a node. This is done to reduce the errors that may occur because of poor scanning, noise etc. Because of the page limitation of this conference we could not accommodate complete identification scheme here. However, a brief of the identification technique is proposed as follows.

We use pitch feature at the beginning of the tree. At first using character pitch features, Japanese text portion is identified. After using pitch features, we check whether left reservoir is present in a character or not. If left reservoir is present then the character will be identified as Japanese if any one of the following properties satisfies.

- (i) If the position of water flow level of left reservoir does not coincide with the left boundary of the character.
- (ii) If the position of water flow level of left reservoir coincides with the characters left boundary, and right reservoir exists and water flow level of the right reservoir does not coincide with character right boundary.
- (iii) If left reservoir present but right reservoir is absent, and maximum width of the left reservoir appears at the position of the water flow level.

By the above conditions English (Japanese) characters having left reservoir will be identified as English (Japanese). For example see Fig.10, where some English and Japanese characters are shown to illustrate the above properties. Here English character 'Z' will be identified as English based on the above property (i) and the character 'a' will be identified by property (iii).

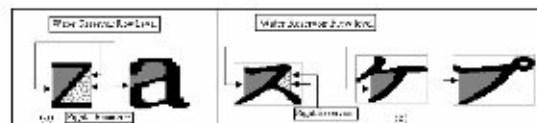
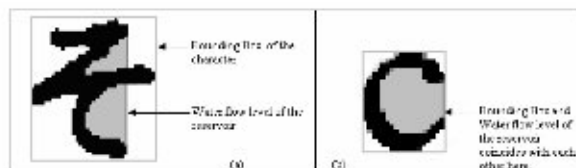


Fig.10. Examples of English and Japanese characters identification based on left reservoirs. (a) English characters (b) Japanese characters.

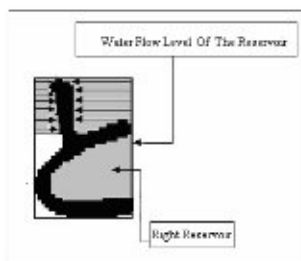
If we consider water reservoir from right side then in some of the English characters like C, E, F, K, Q, R, S, X, Z, e, f, g, k, s, x, z etc. we will get right reservoir. Also there are some Japanese characters where water reservoir from right can be obtained. But we can see there are distinct differences in the properties of the reservoirs obtained in these two scripts. For English characters the water flow level of right reservoir and character right boundary line are same. But in Japanese scripts this is generally not true. For example see Fig.11.

There are very few Japanese characters where water flow level of right reservoir coincides with the right boundary of the character, and these Japanese characters may wrongly be identified as English. For example see Fig.12. To identify such Japanese characters we use profile

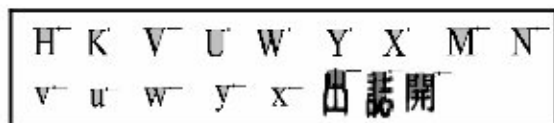
feature at the upper part of the reservoir. We can notice that the maximum column width of the character portion between left and right profiles is similar to the stroke width of the character. Based on this criterion such Japanese characters are identified.



**Fig.11. Relation between character boundary and water flow level of (a) Japanese (b) English character.**



**Fig.12. Water reservoir and profile feature are shown in a Japanese character. Left and right profile is shown only in the upper part of the character.**



**Fig.13. Top reservoirs are shown in some English & Japanese characters. Arrow shows water flow level.**

Top reservoirs and their water flow levels also help us for identification of English and Japanese scripts. If we consider top reservoir then we can see that in most of the English character water flow level passes through the top of the characters. For example see Fig.13. But there are some Japanese characters where water flow level of the top reservoir may pass through the top of the character (Some of them are shown in Fig.13). To identify such English and Japanese characters we check difference of character width and reservoir width. If this difference is less than 2.5 times of the stroke width then it is English, otherwise Japanese.

Sometimes in English we may get some characters where top reservoir flow level may not coincide with top boundary of the characters. See Fig.14 for illustration. Here top reservoir on two English characters 's' and 'e' and two Japanese characters are shown. To identify those English characters from Japanese we check the portion above the reservoir. In English characters upper portion of the top reservoir is not open whereas it is open for Japanese characters. Here upper part of reservoir is marked by arrow to show whether upper part is open or not. We

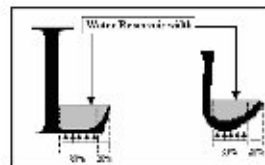
use this distinct property for identification of such English characters. For a character we test this property as follows. From every column of the reservoir's top row, we move upwards to check whether we can reach top boundary of the character without hitting any black pixel or not. If at least in 30% of the columns we can reach the top boundary without hitting black pixel then we say upper part of the character is open. English character 'L' may be identified as Japanese by this property and its identification technique is discussed as follows.

In Fig.15 two similar characters are shown. Here left character is English and right character is Japanese. To identify these two characters we compute width of the top reservoir obtained in this characters. We divide the reservoir width into two parts of which length of left part is 80% and that of right part is 20%. Respective character component in these two portions are noted. We find bottom profile of the left portion of the character and profile distances are computed in this portion. If the range (maximum value - minimum value) of such profile distances is less than the stroke width of the character then we identify that character as English. Otherwise, it is Japanese.

To take care of minor differences between the characters of Japanese and English scripts we use the above tree-base classification scheme for the purpose.



**Fig.14. Identification procedure of some of the English and Japanese characters considering upper part of top reservoir.**



**Fig.15. Identification of two similar shaped characters.**

## 5. Results and discussions

**Dataset:** For experiment 11304 printed characters (8838 Japanese and 2466 English characters) were considered from different mix-documents containing both Japanese and English characters. These characters are collected from advertisement, Windows manual, different magazine like Magazine (ASCII), Magazine (Nikkei Byte), Magazine (Interface) etc.

**Identification accuracy:** We obtained 98.84% (98.62%) identification accuracy on Japanese (English) text. Since there is no ground truth of the data the results are computed manually. Details results are given in Table 3. From the experiment we noticed that 1.16% cases



Japanese characters mis-recognized as English, and 1.38% cases English characters mis-recognized as Japanese. An example of a script identification results on a document image is shown in Fig.16. Here the portions marked by deep horizontal lines were identified as Japanese text by the features based on pitch information. The portions marked by light horizontal line were identified as Japanese text by the other features used in the proposed scheme. Unmarked portions were identified as English text.

**Table.3. Script identification accuracy**

Script (Data size)	Identification accuracy	
	Japanese	English
Japanese (8838)	8736 (98.84%)	102 (1.16%)
English (2466)	34 (1.38%)	2432 (98.62%)



**Fig.16. Identification results of our proposed system. (portions marked by horizontal line segments are identified as Japanese text, and rest are English).**

**Size invariance:** The proposed scheme does not depend on the size of characters in a text line. Also the propose approach is font and case insensitive. The use of simple features, which are easy to compute, makes our system fast.

**Table 4: Comparison of results**

Method	Script used	Classifier used	Accuracy obtained
Gllavata and Freisleben [10]	Ideografic-English	K-NN	89.10%
Dhanya et al. [9]	Tamil-English	SVM	96.03%
Jaeger et al. [11]	Arabic-English	SVM	90.93%
	Chinese-English	SVM	93.43%
	Korean-English	K-NN	94.04%
	Hindi-English	K-NN	97.51%
Proposed method	Japanese-English	Tree-classier	98.79%

**Error Analysis:** Major errors of the system were incurred mainly because of improper character segmentation. This improper character segmentation was due to touching of two or more consecutive characters or splitting of one character into two parts. For example, see the numerals '800' marked by square boxes in the 5<sup>th</sup> line of Fig.16. Here the English numeral string '800' is identified as Japanese and this is because of touching of

there numerals '8' '0' '0' as a single string. To get an idea about splitting errors see the Japanese character marked by thick rectangular box in the last line of Fig.16. Here the Japanese character has been segmented into two components and hence the error occurred. At present we do not have any rejection criteria. In future we plan to implement it.

**Comparison of results:** Since this is the first work on Japanese-English script recognition from a single text line, we cannot compare our results. However to get an idea about the identification accuracy with other existing pieces of work on script identification, some comparison results are given in Table 4. From the Table it can be seen that our proposed method gives better results than that of other published work.

## 6. Conclusion

Script separation is very useful for multi-lingual OCR development. In this paper, an automatic technique for identification of Japanese and English scripts from a single line of a document page is proposed. Based on 11304 data, we obtained overall 98.79% accuracy from the experiment. To the best of our knowledge the proposed work on the identification of English/Japanese text from a single line is the first work of its kind.

## References

1. A. L. Spitz, "Determination of the script and language content of document images", IEEE Trans. on PAMI, vol. 19, pp. 235-245, 1997.
2. 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, pp. 176-181, 1997.
3. J. Ding, L. Lam and C. Y. Suen, "Classification of oriental and European scripts by using characteristic features", In Proc. 4th ICDAR, pp. 1023-1027, 1997.
4. T. N. Tan, "Rotation invariant texture features and their use in automatic script identification", IEEE Trans. on PAMI, vol. 20, pp. 751-756, 1998.
5. U. Pal, S. Sinha and B. B. Chaudhuri, "Multi-Script Line identification from Indian documents", In Proc. 7<sup>th</sup> ICDAR, pp.880-884, 2003.
6. U. Pal, A. Belaid and Ch. Choisy, "Touching numeral segmentation using water reservoir concept", Pattern Recognition Letters, vol.24, pp. 261-272, 2003.
7. B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", Pattern Recognition, vol. 31, pp. 531-549, 1998.
8. W. M. Pan, C. Y. Suen and T. D. Bui, "Script identification using steerable Gabor filters", In Proc. ICDAR-2005, pp.883-887, 2005.
9. D. Dhanya, A. G. Ramakrishna and P. B. Pati, "Script identification in printed bilingual documents", Sadhana, vol.27, pp.73-82, 2002.
10. Gllavata and B. Freisleben "Script recognition in images with complex backgrounds". Int. Symposium on Signal Processing and Information Technology, pp.589-594, 2005.
11. S. Jaeger, H. Ma, and D. Doermann, "Identifying script on word-Level with informational confidence", In Proc. 8<sup>th</sup> ICDAR, pp.416-420, 2005.