International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, July-august. 2013 pp-2232-2238 ISSN: 2249-6645

An Empirical Study on Identification of Strokes and their Significance in Script Identification

Sirisha Badhika
*Research Scholar, Computer Science Department, Shri Jagdish Prasad Jhabarmal Tibrewala University, India

***ABSTRACT**: India is a multilingual, multi-script country. There are totally 22 official languages and 12 scripts in India. People adopted to use two or more languages resulting in bilingual and trilingual documents. Many official documents are available with a combination of local language, English and sometimes Hindi. In this context script identification relies on the fact that each script has unique spatial distribution and visual attributes that make it possible to distinguish it from other scripts. Many script identification methods such as Distribution of an index of optical density method, identification of frequently occurring connected component templates, filtered pixel projection profiles vertical and horizontal projection profiles of document images were proposed earlier. In this work, a simple technique for script identification from a set of English, Telugu, and Devanagari document images in printed form is presented. The proposed system uses stroke features, pixel distribution along a sequence of words.*

***KEYWORDS:** strokes, vertical projection, Horizontal projection, script identification, peaks*

I. INTRODUCTION

Script identification is an important topic in pattern recognition and image processing based automatic document analysis and recognition. The objective of script identification is to translate human identifiable documents to machine identifiable codes. Script identification may seem to be an elementary and simple issue for humans in the real world but it is difficult for a machine, primarily because different scripts (a script could be a common medium for different languages) are made up of different shaped patterns to produce different character sets. OCR is of special significance for a multi-lingual country like India, where the text portion of the document usually contains information in more than one language. The official languages of India are Assamese, Bangla, (Bengali) English, Gujarati, Hindi, Kankanai, Kannada, Kashmiri, Malayalam, Marathi, Nepali, Oriya, Panjabi, Rajasthani, Sanskrit, Tamil, Telugu and Urdu. Of them, Devanagari script is used to write Hindi, Marathi, Rajasthani, Sanskrit and Nepali language while Bangla script is used to write Assamese and Bangla (Bengali) languages. The script of Hindi is Devanagari (which is also used to write Nepali, Marathi and Sindhi), while that of Bangla is called Bangla (also used to write Assamese and Manipuri). There is a strong structural similarity between Urdu and Arabic, the third most popular language in the world. Hindi and Bangla are the fourth and fifth most popular languages in the world respectively. Indian scripts differ from one another significantly. Most of the Indic scripts belong to the family of "syllabic alphabets" and include symbols to represent vowels (V), consonants (C), and vowel modifiers (M) for nasalization of vowels and consonants. A consonant that does not contain the implicit vowel sound is sometimes termed as a half-consonant (C'). Vowel symbols combine with consonants in the form of diacritical marks known as matras. A "character" in an Indic script refers to the orthographic syllabic unit [1]. Syllabic means that text is written using consonants and vowels that together form syllables. From the angle of language specificity, a word is derived from these syllables. In certain cases one syllable forms the complete word and in other cases the number of syllables in a word can be extended. Some scripts, like Hindi, Bengali and Assamese have horizontal and vertical linear features, while others like Telugu, Tamil and Malayalam have complicated curves. Many characters in Bangla and Devanagari script have a horizontal line at the upper part. Different Indian scripts also have different textural properties. Devanagari characters exhibit two-dimensional nature (Figure. 1) and thus the absolute positions of the strokes within the character or their relative positions with respect to the base consonant are generally regarded as important information for recognition.

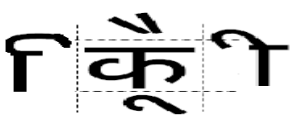

Figure 1: Two-dimensional structure: some possible matras for a Devanagari consonant

Generally human system identifies the script in a document using some visible characteristic features such as horizontal lines, vertical lines, strokes which are visually perceivable and appeal to visual sensation. Our present work is concerned with script separation and not the language separation. We are proposing to use vertical and horizontal projection profiles of document images for determining scripts in machine generated documents. Projection profiles of document images are sufficient to characterize different scripts at page level. The current paper uses horizontal and vertical projection profile features, stroke features for printed Devanagari, Telugu and English script.

II. LITERATURE REVIEW

Identification of stokes and their positions are considered as important information for online recognition of handwritten characters and words in oriental and Indic family of scripts especially because of their multi-stroke and two-dimensional nature. The significance of stroke size and position information for Devanagari word recognition by means of an empirical evaluation of three different word pre-processing schemes. These schemes involved retaining different degrees of stroke size and position information from the original input word. The experiments show that it is indeed possible to reliably recognize a handwritten Devanagari word written as discrete symbols, even in the absence of any size and position information [1]. Script recognition [2] by identifying stokes in document image segmentation were presented by identifying the valleys of the horizontal projection profiles and the position between two consecutive horizontal projections denotes the boundary of a text line. Using these boundary lines, document image is segmented into several text lines. Each text line is further segmented into several text words using the valleys of the vertical projection profile. To recognize online handwritten Gurumukhi words [3] a new step of rearrangement of recognized strokes in online handwriting recognition procedure were presented by classifying recognized strokes as dependent and major dependent strokes, and the rearrangement of strokes with respect to their positions. The combination of strokes to recognize character has achieved an overall recognition rate as 81.02% in online handwritten cursi handwriting for a set of 2576 Gurumukhi dictionary words. The script-line identification techniques in [4], [5] were modified in [6] for script-word separation in printed Indian multi-script documents by including some new features, in addition to the features considered earlier. The features used are headline feature, distribution of vertical strokes, water reservoir-based features, shift below headline, left and right profiles, deviation feature, loop, tick feature and left inclination feature. Tick feature refers to the distinct “tick” like structure, called “telakattu”, present at the top of many Telugu characters. This helps in separating Telugu script from other scripts. The vertical projection profile (or vertical histogram) of a print line consists of a simple running count of the black pixels in each column of that line. Baird et. al., [8] used the projection profile to horizontally segment characters and improved on this further by applying second order derivative on this profile. The resultant peak along with a threshold signifies in determination of the segmentation boundaries. Lu [9] designed a peak-to-valley function based on the ratio of sum of the differences between minimum value and the peaks on each side obtained from the second – order difference profile. This ratio exhibits low valleys with high peaks on both sides.

One early attempt to characterize script of a document without actually analyzing the structure of its constituent connected components was made by Wood et al. They proposed to use vertical and horizontal projection profiles of document images for determining scripts in machine generated documents. They argued that the projection profiles of document images are sufficient to characterize different scripts. For example, Roman script shows dominant peaks at the top and bottom of the horizontal projection profile, while Cyrillic script has a dominant midline and Arabic script has a strong baseline. On the other hand, Korean characters usually have a peak on the left of the vertical projection profile. However, the authors did not suggest how these projection profiles can be analysed automatically for script determination without any user intervention. Also, they did not present any recognition result to substantiate their argument [7]. Liang and others [10] improved the filtering to accommodate touching characters. They not only used the projection profile, an algorithm is proposed which used the differences between the upper and lower projection profiles of the script line are adapted for segmentation.

III. SCRIPT FEATURES

Every script defines a finite set of text patterns called alphabets. Alphabets of one script are grouped together giving meaningful text information in the form of a word, a text line or a paragraph. Thus, when the alphabets of the same script are combined together to yield meaningful text information, the text portion of the individual script exhibits a distinct visual appearance. The distinct visual appearance of every script is due to the presence of the segments like- horizontal lines, vertical lines, upward curves, download curves, descendants and so on. The presence of such segments in a particular script is used as visual clues for a human to identify the type of even the unfamiliar script. In most Indian languages, a text line may be partitioned into three zones. The upper-zone denotes the portion above the head-line, the middle zone covers the portion below head-line and the lower-zone is the portion below base-line. For the text having no head-line, the mean-line separates upper-zone and middle-zone. An imaginary line, where most of the uppermost (lowermost) points of characters of a text line lie, is called as mean- line (base-line). Examples of zoning are shown for English and Devanagari scripts are given below in fig 2(a &b).

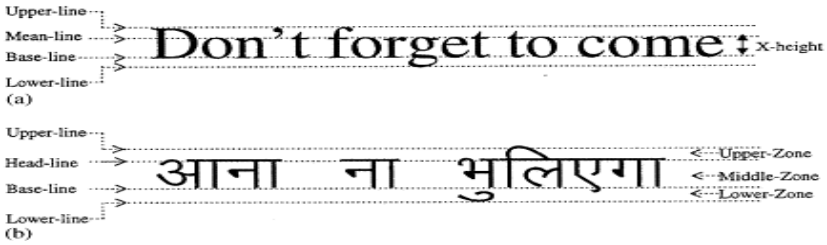

Figure 2: Text zone separation for English and Devanagari script

3.1 PRE-PROCESSING: After scanning the document, the document image is subjected to pre-processing for background noise elimination, skew correction and binarization to generate the bit map image of the text is necessary but in this project input images created saved as a bit map image. The pre-processed image is then subjected to feature extraction. Any language identification method requires conditioned image input of the document, which implies that the document should be noise free and skew free. Apart from these, some recognition techniques require that the document image should be segmented, thresholded and thinned. All these methods, help in obtaining appropriate features for text language identification processes. The pre-processing techniques such as noise removal and skew correction are not necessary for the data sets that are manually constructed by downloading the documents from the Internet.

3.2 FEATURE EXTRACTION: Projection profiles have been used extensively in the field of document analysis especially in skew removal and for block classification.

a. HORIZONTAL PROJECTION PROFILE: The horizontal projection profile of the document image and vertical white spaces are used to compute the separation between lines. First the number of columns and rows are computed for the document image in pixel count as i and j pixels. The horizontal projection is represented by equation (3.1) given below:

$$M \quad H[i] = \sum_{j=1}^{m} f[i, j] \quad - \quad (3.1)$$

Where m = number of pixels in the y direction i.e. vertically
i = Represent the row number.
j = Represent the column number.

In the binary image of each text line, count the number of black pixels in each row and take the count. This gives the horizontal projection profile of that image. Horizontal projection profile for English and Devanagari scripts are shown below. Horizontal projection for sample English script is presented in fig 3.

Once the range is selected can modify the formatting such as font size alignment including vertical alignment font color number formats borders background and so on. To access these settings choose format cells from the main menu bar or right click and choose Format Cells from the pop-up menu. This command opens the dialog box shown in Figure. If the text does not fit the width of the cell can increase the width by hovering the mouse over the line separating two columns until the cursor changes to a double arrow then click the left button.

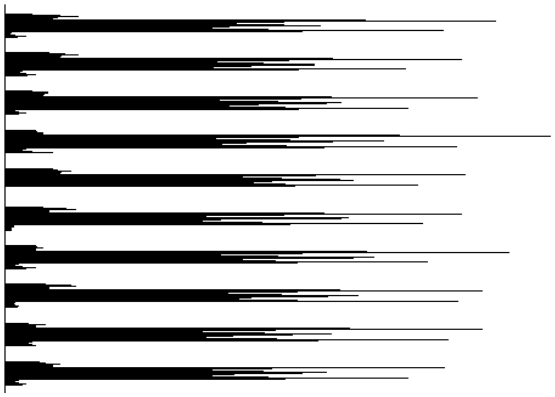

Figure 3: Horizontal projection profile of English script

Features calculated for this image are given below.
Number of peaks= 13
Number of valleys= 56
Number of Strokes= 22.6

Algorithm to calculate Peaks and Valleys:
1.Read the image
2.Convert the rgb image to binary image.
3.Count the number of black pixels in each row and obtain the vector of total image.
4.Normalize the vector
5.Then calculate the mean, maximum, minimum values for the normalized vector.
6.Calculate the peak vector and valleys vector as given below.

If there are continuous one's in a row greater than the horizontal maximum threshold value (horizontal threshold value is calculated for each text line. horizontal threshold value = 50% of the difference between maximum value and mean value), then such continuous one's are retained resulting in peaks. And, if there are continuous one's in a row less than the horizontal minimum threshold value (horizontal threshold value is calculated for each text line. Horizontal threshold value = 50% of the difference between mean value and minimum value), then such continuous one's are retained resulting in valleys.

Figure 4: Mean, Maximum, Minimum

b. VERTICAL PROJECTION PROFILE: Similarly the vertical projection profile of the document image and horizontal white spaces are used to compute the separation between words. First the number of columns and rows are computed for the document image in pixel count as i and j pixels. The vertical projection is represented by equation (3.2) given below:

$$H[i] = \sum_{j=1}^{m} f[i, j] \quad - \quad (3.2)$$

Where m = number of pixels in the x direction i.e. horizontally
i = Represent the column number.
j = Represent the row number.

The computation of the difference profile from the projection profile H For every entry in H starting from index 2 is presented by equation (3.3)

$$D[i] = H(i-1) - H(i) \quad - \quad 3.3$$

Where i = current element under evaluation between the range of
2: size of H

In the binary image of each text line, count the number of black pixels in each column of each row and take the count. This gives the vertical projection profile of that image. Vertical projection for sample English script is presented in fig 5.

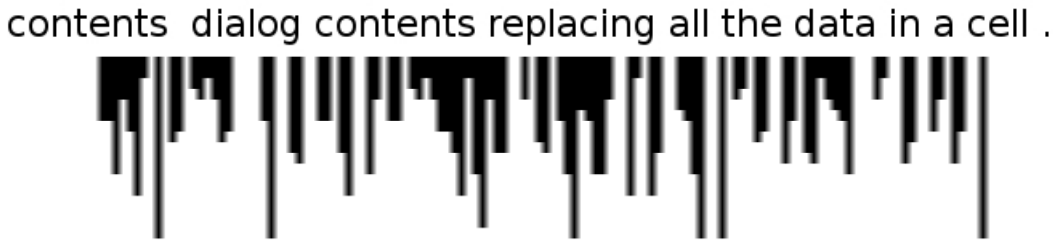

Figure 5: Vertical projection profile for English script

Features calculated for this image are given below.
Number of peaks= 13
Number of valleys= 60
Number of Strokes= 28
Stroke length= 478

Algorithm to calculate Strokes and stroke length:

1.Read an image
2.Convert rgb image to binary image
3.Get the top and bottom of the each text line in an image using vertical projection profile.
4.Measure the height of each text line.
5.For each and every row count the number of black pixels vertically. This gives the vertical projection profile vector.
6.Normalize the vector.
7.Then calculate the strokes and stroke lengths as given below.
If there are continuous one's in a column greater than the vertical threshold value (vertical thres hold value is calculated for each text line. Vertical threshold value = 75% of the X-height of that text line), then such continuous one's are retained resulting in a strokes.
8.Count the number of strokes, measure stroke lengths.

IV. RESULTS/FINDINGS

Three test sample images downloaded from internet (Google, Wikipedia for Hindi, Telugu and English) and the test sample values are given below.

Test Sample – 1:

इन व्यंजनों के हर प्रकार में पकवानों का एक अच्छा खासा विन्यास और पकाने के कई तरीकों का प्रयोग होता है यद्यपि पारंपरिक भारितीय भोजन महत्वपूर्ण हिस्सा शाकाहारी है लेकिन कई परम्परागत भारतीय पकवानों में मुर्गा बकरी भेड़ का बच्चा मछली और अन्य तरह के मांस भी शामिल हैं भोजन भारतीय संस्कृति का एक महत्वपूर्ण हिस्सा है जो रोज़मर्रा के साथ साथ त्योहारों में भी एक महत्वपूर्ण भूमिका अदा करता है कई परिवारों में हर रोज़ का मुख्य भोजन दो से तीन दौर में कई तरह की चटनी और अचार के सा रोटी और चावल के रूप में कार्बोहाइड्रेट के बड़े अंश के साथ मिष्ठान सहित लिया जाता है भोजन एक भारतीय परिवार के लिए सिर्फ खाने के तौर पर ही नहीं बल्कि कई परिवारों के एक साथ एकत्रित होने सामाजिक संसर्ग बढाने लिए भी महत्वपूर्ण है

Figure 6: Test Sample 1 of Devanagari Script

Four features (no. of strokes, stroke lengths, no. of peaks, no. of valleys) are calculated the values are tabulated in the below table (4.1):

Table 4.1: Features Values for the Test Sample - 1 (Devanagari script)

No. of strokes	Stroke length	No. of Peaks	No. of Valleys
1.2	21.6	9	56.7

Test Sample – 2:

Age 26, lean, hard, the consummate loner. On the surface he appears good-looking, even handsome; he has a quiet steady look and a disarming smile which flashes from nowhere, lighting up his whole face. But behind that smile, around his dark eyes, in his gaunt cheeks, one can see the ominous stains caused by a life of private fear, emptiness and loneliness. He seems to have wandered in from a land where it is always cold,

Figure 7: Test Sample 2 of English Script

Four features (no. of strokes, stroke lengths, no. of peaks, no. of valleys) are calculated the values are Tabulated in the below table (4.2).

Table 4.2: Features values for the test sample-2 (English script)

No. of strokes	Stroke length	No. of Peaks	No. of Valleys
24.9	428	13	57

TEST SAMPLE – 3:

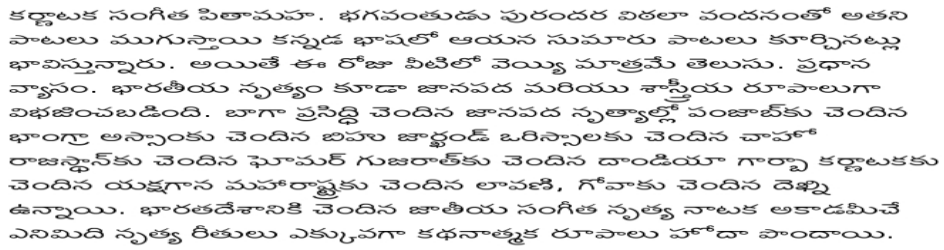

Figure 8: Test Sample 3 of Telugu Script

Four features (no. of strokes, stroke lengths, no. of peaks, no. of valleys) are calculated the values are tabulated in the below table (4.3).

Table 4.3: Features values for the test sample-3(Telugu script)

No. of strokes	Stroke length	No. of Peaks	No. of Valleys
0	**0**	**12.8**	**54**

4.1 OBSERVATION:

Devanagari*:* It is observed that for Devanagari script the number of strokes vary between a minimum of 1 to 6. As the number of strokes vary so do the stroke length. It varies between 54 -59. The peak value is almost constant and is always in the vicinity of 9.

English: *The* average number of strokes for English is always greater than 20 which is unique to this script. The average stroke length is much greater than all other scripts under consideration and is greater than 400. This is due to the fact that English script is having more vertical line like structure characters. Peaks and Valleys for English are always constant and are 13 and 57 respectively.

***Telugu*:** The no. of strokes and the stroke length are almost 0. Because vertical line like structure character are very less in Telugu script. The peak value is a constant with a value of 12 and the valley has an average value of 52.

Script identification: It is observed, if a test image after converted into black and white and calculate all the 4 features (no. of strokes, stroke lengths, no. of peaks, no. of valleys) and they are then compared with the training data base. And for each parameter, the script with minimum distance is identified and coded into a 1x4 vector.

- If more than two elements of the minimum distance vector-V has a same value X, then the test image script is identified as the script with code X. This condition is useful for identifying English, Telugu.
- If third element is either 2 or 4 and if remaining elements are not equal to 4 then the test image script is identified as the script with code 2. (This condition occurs only for Devanagari script.)

V. CONCLUSIONS

In this work a method to identify the document images of English, Hindi and Telugu scripts from a document image set is presented. The approach is based on the analysis of the horizontal projection profile and vertical projection profile of the document images and explores the features like strokes, peaks and valleys of pixel distribution.

In this work we observed that, vertical stroke features are efficient to classify south Indian languages from north Indian languages. To classify north Indian languages among themselves and also south Indian languages among themselves pixel distribution is used. To further improve the efficiency of classification and to cover even more scripts, additional features like entropy and energy distribution can be explored as a future task.

VI. ACKNOWLEDGMENT

I would like to record my sincere thanks to my research supervisor **Dr. L. PRATAP REDDY,** Director R&D Cell, Prof. in ECE Department, JNTUH College of Engineering and Mrs. Karunasri JNTUH College of Engineering. Their vision, breath of knowledge, perseverance and patience has been the motivating factors behind this work.

REFERENCES

[1]. Bharath A. and Sriganesh Madhvanath, "On the Significance of Stroke Size and Position for Online Handwritten Devanagari Word Recognition: An Empirical Study ", 2010 IEEE, International Conference on Pattern Recognition.

[2]. M.C. Padma and P. A. Vijaya, "Script Identification of Text Words from a Tri-Lingual Document Using Voting Technique",2010 International Journal of Image Processing, Volume (4): Issue (1)

[3]. Anuj Sharma, Rajesh Kumar and R.K. Sharma ,"Rearrangement of Recognized Strokes in Online Handwritten Gurumukhi Words Recognition",2009 10th International Conference on Document Analysis and Recognition

[4]. U. Pal and B.B. Chaudhuri, "Identification of Different Script Lines from Multi-script Documents," Image & Vision Computing, vol. 20, no. 13-14, pp. 945-954, Dec. 2002.
[5]. U. Pal, S. Sinha, and B.B. Chaudhuri, "Multi-script Line Identification from Indian Documents," Proc. Int'l Conf. Document Analysis & Recognition, Edinburgh, pp. 880-884, Aug. 2003.
[6]. S. Sinha, U. Pal, and B.B. Chaudhuri, "Word-wise Script Identification from Indian Documents," Lecture Notes in Computer Science: IAPR Int'l Workshop Document Analysis Systems, Florence, LNCS-3163, pp. 310-321, Sep. 2004
[7]. S.L. Wood, X. Yao, K. Krishnamurthy, and L. Dang, "Language Identification for Printed Text Independent of Segmentation," Proc. Int'l Conf. Image Processing, Washington D.C., vol. 3, pp. 428-431, Oct. 1995.
[8]. H.S. Baird, S. Kahan, and T. Pavlidis,"Components of an Omni- font Page Reader," Proc. Eighth Int'l Conf. Pattern Recognition, Paris, pp. 344- 348, 1986.
[9]. Y. Lu, "On the Segmentation of Touching Characters," lnt'l Conf. Document Analysis and Recognition, Tsukuba, Japan, pp. 440-443, Oct. 1993.
[10]. S. Liang, M. Ahmadi, and M. Shridhar, "Segmentation of Touching Characters in Printed Document Recognition," Proc. Int'l Conf. Document Analysis and Recognition, Tsukuba City, Japan, pp. 569-572, Oct. 1993.