Word-level Script Identification for Handwritten *Indic* scripts

Pawan Kumar Singh, Ram Sarkar, Mita Nasipuri
Computer Science & Engineering Department
Jadavpur University
Kolkata, India
{pawansingh.ju, raamsarkar, mitanasipuri}@gmail.com

David Doermann
Institute for Advanced Computer Studies
University of Maryland, College Park, USA
doermann@umiacs.umd.ed

Abstract— **Automatic script identification from handwritten document images facilitates many important applications such as indexing, sorting and triage. A given Optical Character Recognition (OCR) system is typically trained on only a single script but for documents or collections containing different scripts, there must be some way to automatically identify the script prior to OCR. For *Indic* script research, some results have been reported in the literature but the task is far from solved. In this paper, we propose a word-level script identification technique for six handwritten *Indic* scripts- *Bangla*, *Devanagari*, *Gurumukhi*, *Malayalam*, *Oriya Telugu* and the *Roman* script. A set of 82 features has been designed using a combination of *elliptical* and *polygonal approximation* techniques. Our approach has been evaluated on a dataset of 7000 handwritten text words, using multiple classifiers. A Multi-Layer Perceptron (MLP) classifier was found to be the best classifier resulting in 95.35% accuracy. The result is progressive considering the complexities and shape variations of the *Indic* scripts.**

Keywords*—— Script Identification, Indic scripts, Handwritten words, Elliptical based features, Polygonal approximation based features, Multiple Classifiers.

I. Introduction

OCR techniques transform scanned images of handwritten, typewritten or printed text into machine-encoded text. In a multilingual country like India, a text document may be written in more than one language and/or more than one script. OCR of such document pages can be designed to fit into one of the following two categories: (1) a generalized OMNI-Script recognition system which can recognize characters all possible scripts present on the document page, or (2) a script identification scheme to identify the script, followed by script specific recognition. In general, recognition of the characters written in different scripts by a single OCR module is much more difficult. This is because the features, which are necessary for the character recognition, depend heavily on the structural properties, styles and the nature of the writings or fonts which largely differs from one to another [1]. For example, features used for recognizing *Devanagari* script characters might not be useful for identifying *Roman* script. The second option for handling documents in a multi-script environment is to use a bank of OCR engines corresponding to different scripts which tend to be more reliable. However, it requires a priori knowledge of the script in which the document is written. Unfortunately, this information may not be readily available. Manual identification of the documents' scripts may be monotonous and time consuming, therefore, it is necessary to identify the script of the input document before feeding the document to the corresponding OCR system. The ability to reliably identify the script using the least amount of textual data is essential when dealing with document pages that contain text words of different scripts [2-3]. Additionally, identifying handwritten scripts at word-level is much more challenging due to the fact that the features from a few characters constituting the word may not be sufficient at times for most script recognition approaches.

Several works have been reported in literature for script identification of both *Indic* and non-*Indic* scripts. Spitz [2] proposed a method for distinguishing between *Asian* and *European* languages by examining the upward concavities of connected components. Tan [3] developed a rotation invariant texture feature extraction method for automatic script identification for six languages-*Chinese*, *Greek*, *English*, *Russian*, *Persian* and *Malayalam*. Hochberg *et al.* [4] presented a system to automatically identify six different scripts-*Arabic*, *Chinese*, *Cyrillic*, *Devanagari*, *Japanese* and *Roman.* In this work, a set of five features was extracted from all of the connected components assuming eight-connectedness and was used to train and test a Linear Discriminant Analysis (LDA) classifier. In the context of *Indic* scripts, most of the published methodologies [5-8] focused on printed text, but some work [9-13] has been applied to handwritten documents. Zhu *et al.* [9] proposed a novel approach to language identification for documents containing mixture of handwritten and printed text written in eight languages- *Arabic*, *Chinese*, *English*, *Hindi*, *Japanese*, *Korean*, *Russian* and *Thai* using image descriptors constructed from a codebook of shape features. Jain *et al.* [10] presented a new method for writer identification based on contour gradients to capture local shape and curvature, with character segmentation to create a pseudo-alphabet for a given handwriting sample. This approach is taken by forensic document examiners. Sarkar *et al.* [11] proposed eight holistic features for word-level script identification from *Bangla* and *Devanagari* handwritten texts mixed with *Roman* script by using a MLP classifier. Singh *et al.* [12] reported an intelligent feature based technique for word-level script identification of *Devanagari* mixed with *Roman* script using a combination of topology and convex hull based features. For selection of a suitable classifier, the technique [12] was tested using multiple

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

classifiers using the statistical significance tests described in [13]. Singh *et al.* [14] described a robust word-wise script identification scheme for five handwritten scripts- *Bangla, Devanagari, Malayalam, Telugu* and *Roman*. In addition to the above mentioned research contributions, no other significant work on word-level script identification from handwritten *Indic* script documents could be found in the literature. In a multilingual country like India with so many scripts, this is truly a significant limitation. This is the primary motivation behind the development of the word-level script identification technique for six popular handwritten *Indic* scripts - *Bangla*, *Devanagari*, *Gurumukhi*, *Malayalam*, *Oriya* and *Telugu* along with *Roman* script.

II. PROPOSED APPROACH

The first step in our approach is to provide feature extraction and this is accomplished by using a novel combination of *elliptical* and *polygonal approximation* based features which are described in the following subsections:

A. *Elliptical Based Features*

A set of novel features based on elliptical regions of the word images has been designed for recognizing these scripts. *Elliptical* based features are extracted from the contour and the local regions of a word image so that it is easier to isolate a particular script. For example, in some *Indic* script alphabets (like *Devanagari*, *Bangla* and *Gurumukhi*, etc.) it is noted that many characters have a horizontal line at the upper part called Matra or Shirorekha. By using *elliptical* features, we distinguish the Matra based scripts from non-Matra based scripts (such as *Malayalam*, *Oriya, Telugu*, *Roman*, etc.) by the fact that different scripts have different pixel densities in some explicit zones or regions. This can better be discriminated with the help of an ellipse which is a closed curved shape like a circle. Moreover, an ellipse can be elongated so that the sum of distances from any two points inside the ellipse (called foci) is always constant. The word images are, in general, elongated in nature which can better covered by an ellipse.

There are certain unknown parameters for an ellipse that one would like to determine: the center (C_x, C_y), the major (a) and minor axes (b), and the orientation (θ), as shown in Fig. 1.

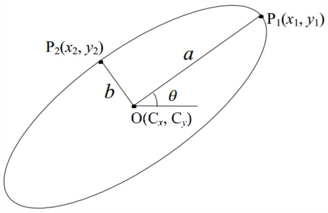

Fig. 1. Illustration of an ellipse with its parameters.

In order to find these parameters, our approach consists of two phases: Identifying the minimum bounding box of a word image and then estimation of these parameters. First, the word image is scanned to locate the bounding box which would completely inscribe an ellipse. Then, for each bounding box, the boundary of an ellipse is traced to find the location of the major axis and its normal i.e. the minor axis. Finally, the orientation of each ellipse is computed so that the features values will be accurately estimated irrespective the skew of the word image. Let *etop*, *ebottom*, *eleft* and *eright* denote “top”, “bottom”, “left” and “right” positions of an ellipse respectively. The center position (C_x, C_y) of the ellipse can be obtained as:

$$C_x = \frac{eleft + eright}{2} \qquad (1)$$

$$C_y = \frac{etop + ebottom}{2} \qquad (2)$$

The periphery of the ellipse within the bounding box is then traced to find the position which maximizes the distance to the center position. This distance is taken to be the major axis. In order to reduce the search time, the rough orientation of the ellipse is estimated by finding the position of the foreground pixel(s) in the first scan line (horizontal). If the number of pixel(s) on the right-hand side is greater than that on the left-hand side, the trace will be in a counterclockwise direction. Otherwise, the trace will be in a clockwise direction. If the number on both the sides is the same, then either side can be used for tracing. Once the search direction is decided, the trace procedure can be performed. The starting position is the first foreground pixel of the first scan line. During the tracing, the position which has the maximal distance to the center position is recorded. The operation is terminated when *eleft* (*eright*) is reached for trace in a counterclockwise direction (in a clockwise direction). This is how a maximum inscribed ellipse is best fit inside the bounding box of a word image. We will now define two more important notations used in this subsection. The *pixel ratio*, P_r, is defined as the ratio of the number of contour pixels (object) to the number of background pixels and the *pixel count*, P_c, is defined as the number of contour pixels. The features are described in detail:

1) Maximum Inscribed Ellipse: The height and width of the bounding box are calculated for each word image. A representative ellipse is then inscribed (considering the orientation of the ellipse) inside this bounding box having major and minor axes equal to the width and the height of the bounding box and the center of an ellipse is also the center of the corresponding bounding box. This ellipse divides the word image into eight regions R_i, i=1, 2…., 8 which are shown in Fig. 2(a). The bounding box along with the inscribed ellipse for a handwritten *Bangla* word image is shown in Fig. 2(b). Taking the values of P_r from these eight regions, as shown in Fig. 2(a), eight features (F1-F8) for each handwritten word image are estimated. Now, another type of feature, P_c along n (n= 8 for the present work) lines parallel to major/minor axis of the representative ellipse are computed. The *mean* and *standard deviation* of the values of P_c along major/minor axis are taken as four additional features (F9-F12). This has been done for detection of the horizontal (or sometimes vertical) strokes for some of the Matra based scripts and vertical (or sometimes horizontal) strokes for some of the non-Matra based scripts.

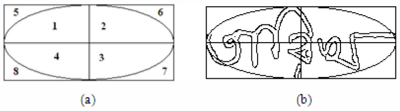

Fig. 2. Illustration of fitting (a) an ellipse inside the minimum boundary box which divides a handwritten *Bangla* word image in eight regions as shown in (b).

2) Sectional inscribed Ellipse: Each of the word image surrounded by the minimum bounding box is again divided into four equal rectangles (see Fig. 3) and a representative ellipse is fit into each of these rectangles using the same procedure as in the previous subsection. This done for two reasons. The first is to get the local information of the word images, and the second is to accurately identify the structure of strokes present in different script words. This ultimately helps us to distinguish between the words written in one script and words written in another script. As a result, every ellipse produces eight regions inside its rectangular area *namely*, R_{ij} where $1 \le i \le 4$ and $1 \le j \le 8$ which makes 8*4=32 regions in total. A total of 32 feature values (F13-F44) using the P_r values are computed from the 32 ellipses in similar fashion. These features have also been considered keeping in mind the presence of ascenders (character components that extend over Matra) and descenders (character components or modified shapes that lie below the actual character) of certain scripts.

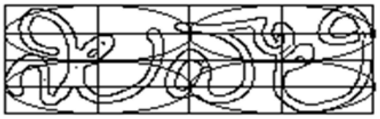

Fig. 3. Diagram showing the four imaginary ellipses inscribed within the four rectangles which is fitted inside the boundary box for a handwritten *Malayalam* word image.

3) Concentric Ellipses: It can be seen from the structural analogy of the *Indic* scripts that some Matra based script words have more diverse pixel density concentrated towards their center which has motivated us to apply the notion of concentric ellipses. These feature values are computed by taking the entire topology of the word image. A primary ellipse is made circumscribing the word image with center taken to be the midpoint of its minimum bounding box. The values of the major and minor axes of the ellipse are taken into consideration. After fitting the primary ellipse, three concentric ellipses are drawn inside the primary ellipse having the same center point as the primary ellipse and semi major and semi minor axes equal to 1/4th, 2/4th and 3/4th of semi major and minor axes of the primary ellipse respectively. These four ellipses divide each of the word images into four regions- R_{e1}, R_{e2}, R_{e3} and R_{e4}. The partitioning of the four regions on a sample handwritten *Devanagari* word image is shown in Fig. 4.

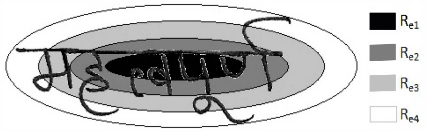

Fig. 4. Figure showing the elliptical partition of four regions on a sample handwritten *Devanagari* word image.

From the four regions, four features values (F45-F48) considering the P_r's and four feature values (F49-F52) considering the P_c's of the regions R_{e1}, R_{e2}, R_{e3} and R_{e4} respectively are estimated. The remaining six features (i.e., F53-F58) are taken as the corresponding differences of the P_r's and P_c's between the regions R_{e1} and R_{e2}, R_{e2} and R_{e3}, R_{e3} and R_{e4} respectively. The *elliptical* based features (F1-F58) are suitably normalized by the height and width of the corresponding word image.

B. *Polygonal Approximation*

Polygonal approximation [15] can be set to ignore the minor variations along the edge, and instead capture the overall shape. This is useful because it reduces the effects of discrete pixelization of the contour which in turn can cause smooth discrimination between the non-Matra script words. In general, there are two methods to realize it, *viz*., *merging* and *splitting*.

1) Distance Threshold Method: Distance threshold method is a merging method of polygonal approximation which is stated as follows:

Step 1: Choose a point as starting point, on the contour of the input word image.
Step 2: Choose another point on the contour and consider it an end point.
Step 3: Construct a line from starting point to the end point.
Step 4: Compute the squared error for every point on the contour from starting point to end point along the segment/line using Eqn. (5).
Step 5: If the error exceeds a threshold value say, μ
(i) Reassign the line end point to the previous point of the current end point and repeat steps 4 and 5.
Otherwise,
(ii) Draw a line from starting point to end point.
Step 6: End point of this line segment is considered as new starting point and repeat step 2 to step 6 until all the contour points are covered.

The value of the threshold (μ) has been taken as three which provides optimal results for most of the cases. The distance $d_k(i, j)$ from curve vertex $V_k = (x_k, y_k)$ to the corresponding approximation linear segments (V_i, V_j) is defined (see Fig. 5) as follows:

$$d_k(i,j) = \frac{|(x_j - x_i)(y_i - y_k) - (x_i - x_k)(y_j - y_i)|}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \qquad (3)$$

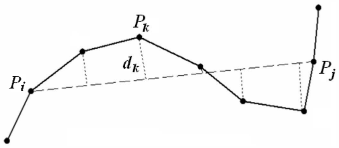

Fig. 5. Measuring the distance from a point P_k on the boundary to a linear segment.

One such application is shown in Fig. 6. With distance threshold method, the value of the distance $d_k(i, j)$ for those vertices of the input curve for each line which has the maximum distance from its line segments are stored. With these values, a 12-bin histogram is generated. Each intervals has the number of pixels whose $d_k(i, j)$ value belongs to the interval (a_i, b_i) where, $1\leq i \geq 12$. Here, $a_i=(i-1)*0.25$ and $b_i=i*0.25$. After suitable normalization using the total number of line segments, these values are considered as features (F59-F70).

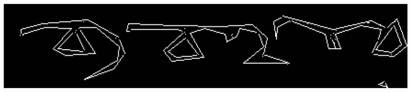

Fig. 6. Illustration of polygon shape using *distance threshold method* for a handwritten *Bangla* word image (allowing squared error d_k up to 3).

2) Fit and Split Method: The simplest way of splitting method is to approximate the curve using the fit and split method [15]. This method is useful for several reasons including simplification of shapes, noise elimination, absence of dislocation of relevant features and the absence of change in vertices. The procedure of this method is described as follows:

Step 1: Divide the boundary into small segments and fit a line for each segment.
Step 2: Draw a line from any point to its farthest point. Identify the longest line segment that connects two farthest points (see Fig. 7 for more illustration).
Step 3: Compute the perpendicular distance from the boundary to the line segment.
Step 4: Check the perpendicular distance with a threshold. If the threshold is crossed, then the line is split.
Step 5: Identify the vertices that are very close to the prominent inflection points of the curve. An inflection point is a point on a curve where the curvature changes its sign.
Step 6: Connect the points to get the approximate polygons.
Step 7: Repeat the process for the new line until there is no longer a need for splitting.

Feature values F71-F82 are estimated using the same rule which is applied to distance threshold method.

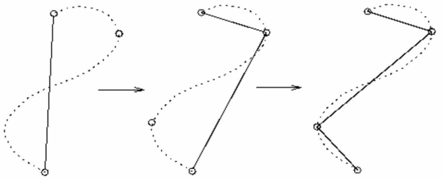

Fig. 7. Illustration of *fit and split* method for polygonal approximation.

III. EXPERIMENTAL RESULTS AND DISCUSSION

For evaluating the word-level script identification technique, 7000 words have been cropped manually from the document pages of the previously mentioned seven handwritten scripts where each script contributes 1000 words. A total of 4200 words (600 words per script) have been used for the training and the remaining 2800 words (400 words per script) have been used for testing the system. The original word images are in grayscale and digitized at 300 dpi. Otsu's global thresholding approach is used to convert them into binary images. The binarized word images may contain noisy pixels which have been removed by using Gaussian filter. Canny Edge Detection algorithm is then applied for detecting the contour of the binarized word images.

The proposed feature set has been individually applied to seven well-known classifiers *namely*- Naïve Bayes, Bayes Net, MLP, Support Vector Machine (SVM), Random Forest, Bagging and MultiClass Classifier. Performances and corresponding success rates of the said classifiers achieved at 95% confidence level are shown in Table I. It can be seen from Table I that MLP has outperformed the other classifiers. So, MLP classifier has been chosen for detailed evaluation using different cross validations with varied number of epochs. The recognition accuracies achieved by the present technique for the current set up are shown in Table II. The overall accuracy of the technique is found to be 95.35% for 5-fold cross validation of the MLP with epoch size of 1500.

It can be observed from the misclassification analysis that the text words written in *Bangla*, *Devanagari* and *Gurumukhi* scripts are misclassified among each other. As mentioned earlier, these scripts have a horizontal line called "Matra" or "Shirorekha" which runs through the upper part of the text word. As a result, the contours of these word images are sometimes found to be almost identical. On the other hand, some of the text words written in *Telugu* script are misclassified with those of *Malayalam* and *Oriya* due to existence of abrupt spaces in between the characters of a single word image and structural similarity in these scripts i.e., non-presence of this "Matra" or "Shirorekha" feature. Some failure cases of the present technique are shown in Fig. 8.

Different feature sets described in [6], [13] and [14] have been evaluated on the present database also. From the experiment (results are shown in Table III), it was noted that the current set up gives higher script identification accuracy and so it may be concluded that the proposed technique outperforms the previous ones.

TABLE I. Success rates of the proposed word-level script identification technique using seven well-known classifiers.

	Classifiers						
	Naïve Bayes	**Bayes Net**	**MLP**	**SVM**	**Random Forest**	**Bagging**	**MultiClass Classifier**
Success Rate (%)	84.69	88.94	**94.35**	93.82	91.52	85.25	90.03
95% confidence score (%)	90.21	92.55	**98.78**	97.65	95.36	91.16	94.85

TABLE II. Recognition accuracies of script identification technique using MLP classifier (the best one is shaded in grey).

	Success Rate of MLP classifier (%)		
Number of Epochs	**3-fold**	**5-fold**	**7-fold**
500	94.25	93.7	93.39
1000	94.44	94.08	94.52
1500	95.13	**95.35**	94.89

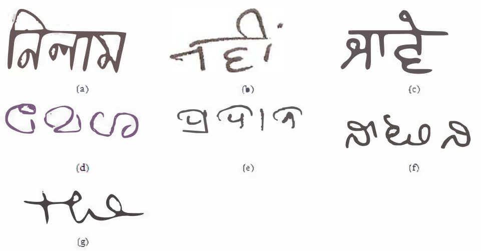

Fig. 8. Sample word images written in: (a) *Bangla*, (b) *Devanagari*, (c) *Gurumukhi*, (d) *Malayalam*, (e) *Oriya*, (f) *Telugu*, and (g) *Roman* scripts are misclassified by the present technique as *Devanagari*, *Gurumukhi*, *Bangla*, *Roman*, *Telugu*, *Oriya*, and *Malayalam* scripts respectively.

TABLE III. Comparison of the present technique with some previous works.

Method	**Feature Set**	**Feature Dimension**	**Scripts used**	**Success Rates (%)**
Ma *et al.* [6]	Gabor Filter	*32*	*Bangla*, *Devanagari*, *Gurumukhi*, *Oriya*, *Malayalam*, *Telugu* and *Roman*.	91.43
Singh *et al.* [13]	Topology and Convex hull based features	*87*		91.07
Singh *et al.* [14]	Shape and Texture based features	*39*		90.25
Proposed Method	*Elliptical* and *polygonal approximation* based features	82		**95.35**

IV. CONCLUSION

Automatic script identification has been a challenging research problem in a multilingual environment addressed by the researchers over the last few decades. In this paper, a word-level script identification technique for six popular handwritten *Indic* scripts - *Bangla*, *Devanagari*, *Gurumukhi*, *Malayalam*, *Oriya, Telugu* along with *Roman* script has been proposed. A combination of *elliptical* and *polygonal approximation* based features has been used for the present work and an accuracy rate of 95.35% has been achieved on a limited dataset. Some of the previously developed techniques have also been evaluated on the current dataset and it is observed that we have achieved better identification precision. The recognition accuracy of our technique is mainly distressed due to presence of noise, improper 'Matra' and poor quality of handwriting. Considering the complexity and variations of the *Indic* scripts, we can claim that we have attained satisfactory results. We will subsequently expand this technique to a larger database containing more number of *Indic* scripts which will consent to a more realistic assessment of the script identification system. The technique could be used as a general word-level script identification module for the development of multi-script OCR system.

REFERENCES

[1] P. K. Singh, "*Script Identification from Multi-Script Handwritten Documents*", M. Tech Thesis, CSE Dept., Jadavpur University, 2013.

[2] L .Spitz, "*Determination of the script and language content of document images*", In: IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, issue 3, pp. 234-245, 1997.

[3] T. N. Tan, "*Rotation invariant texture features and their use in automatic script identification*", In: IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, issue 7, pp. 751-756, 1998.

[4] J. Hochberg, K. Bowers, M. Cannon, P. Kelly, "*Script and language identification for handwritten document images*", In: International Journal on Document Analysis and Recognition, vol. 2, pp. 45-52, 1999.

[5] S. Jaeger, H. Ma, D. Doermann, "*Identifying Script on Word-Level with Informational Confidence*", In: Proc. of 8th International Conference on Document Analysis and Recognition (ICDAR), pp. 416-420, 2005.

[6] H. Ma, D. Doermann, "*Word Level Script Identification on Scanned Document Images*", In: SPIE Conference on Document Recognition and Retrieval, San Jose, CA, pp. 124-135, 2004.

[7] P. B. Pati, A. G. Ramakrishnan, "*Word level multi-script identification*", In: Pattern Recognition Letters, vol. 29, issue 1, pp. 1218-1229, 2008.

[8] P.B. Pati and A.G. Ramakrishnan, "*HVS Inspired System for Script Identification in Indian Multi-script Documents*", In: Lecture Notes in Computer Science: International Workshop Document Analysis Systems, Nelson, LNCS-3872, pp. 380-389, 2006.

[9] G. Zhu, X. Yu, Y. Li, D. Doermann, "*Language Identification for Handwritten Document Images Using A Shape Codebook*", In: Pattern Recognition, vol. 42, issue 12, pp. 3184-3191, Dec. 2009.

[10] R. Jain, D. Doermann, "*Writer Identification Using an Alphabet of Contour Gradient Descriptors*", In: Proc. of 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 550-554, 2013.

[11] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, D. K. Basu, "*Word level script Identification from Bangla and Devanagari Handwritten texts mixed with Roman scripts*", In: Journal of Computing, vol. 2, issue 2, pp. 103-108, 2010.

[12] P. K. Singh, R. Sarkar, N. Das, S. Basu, M. Nasipuri: "*Identification of Devanagari and Roman script from Multiscript Handwritten documents*", In: Proc. of International Conference on Pattern Recognition and Machine Intelligence (PReMI), LNCS 8251, pp. 509-514, Dec 2013.

[13] P. K. Singh, R. Sarkar, N. Das, S. Basu, M. Nasipuri: "*Statistical Comparison of Classifiers for Script Identification from Multi-script Handwritten documents*", In: International Journal of Applied Pattern Recognition (IJAPR), vol. 1, no. 2, pp. 152-172, 2014.

[14] P. K. Singh, A. Mondal, S. Bhowmik, R. Sarkar, M. Nasipuri: "*Word-level Script Identification from Multi-script Handwritten Documents*", In: Proc. of 3rd International Conference on Frontiers in Intelligent Computing Theory and Applications (FICTA), vol. 1, pp. 551-558, 2014.

[15] Y. Mingqiang, K. Kidiyo, R. Joseph, "*A survey of shape feature extraction techniques*", In: Pattern Recognition, Peng-Yeng Yin (Ed.) , pp. 43-90, 2008.