

Using Feature Extraction to Recognize Handwritten Text Image

Khamael A. Khudhair*, Itimad Raheem Ali**

* Al-Nahrain University /College of Science / Computer Department ,Al-jaderyia –Iraq

** AL-Rafidean College / Computer Department

Khamail_abbas@yahoo.com

weffee@yahoo.com

Abstract— In this paper, a robust technique for identifying and recognizing Handwritten Text images is presented. There are many methods for Handwritten Text, but most of them require segmentation or connected component analysis. The Recognition process utilizes the determinant value that produces the features for the Handwritten Text. Image's determinants values are computed by dividing image into blocks then designed Threshold (T) to extract feature, afterwards, use chain code to find the centric point and direction of text. The least square criterion is then utilized to determine the similarity between the existed (in Database file) Handwritten Text with a new query Handwritten Text's images.

Index Terms— Handwritten Text, Feature Extract, Determinant value, chain coding

1 INTRODUCTION

Text in handwritten images typically shows strong variability in appearance due to different writing styles. Appearance differs in the size of the words, slant, skew and stroke thickness. Such variability calls for the development of normalization and pre-processing techniques suitable for recognition of handwritten text. Among the most common preprocessing steps applied in current state-of-the art systems are noise removals, binarization, skew and slant correction, thinning, and baseline normalization [3]. For slant correction, Pastor et al. [17] proposed to use the maximum variance of the pixels in the vertical projection and Vinciarelli et al. [21], observed that non-slanted words show long, continuous strokes. Juan et al. [20] showed that normalizing ascenders and descenders of the text significantly reduce the vertical variability of handwritten images. A linear scaling method applied to whole images has been used in various systems to reduce the overall size variability of images of handwritten text [6, 3, 8]. A drawback of all those approaches is that, they rely on assumptions that may or may not be based upon for a given database. A second drawback is that all those methods are applied to whole images making it difficult to address local changes. Furthermore, the methods for slant correction rely on binarization which, is a non-trivial problem in itself and should be avoided if possible, as Liu et al. [13] found in their benchmark paper. Recently, Espapna-Boquera et al., [7] proposed using trained Multi-Layer-Perceptrons for image cleaning and normalization. While they report competitive results on standard databases, the training and labeling process is time consuming. In contrast to the methods mentioned until now, methods based on image statistics and moments do not suffer from heuristically assumptions and have been extensively studied in the area of isolated digit recognition. Casey [4] proposed that all linear pattern variations can be normal-

ized using second-order moments. Liu et al. [14] used Bi-moment normalization based on quadratic curve fitting and introduced a method to put constrain on the aspect ratio when the x and y axis are normalized independently [12]. Miyoshi et al. [16] reported that computing the moments from the contour of a pattern, and not from the pattern itself, improves the overall recognition results. They do not appear at the same relative location of the letter due to the different proportions in which characters are written by different writers of the language. Even the same person may not always write the same letter with the same proportions.

2-RELATED WORK AND CONTRIBUTION

1.1 Related Work

High accuracy character recognition techniques can provide useful information for segmentation-based handwritten word recognition systems. The research describes neural network-based techniques for segmented character recognition that may be applied to the segmentation and recognition components of an off-line handwritten word recognition system. Two neural architectures along with two different feature extraction Techniques were investigated. A novel technique for character feature extraction is discussed and compared with others in the literature. Recognition results above 80% are reported using characters automatically segmented from the CEDAR benchmark database, as well as standard CEDAR alphanumeric [17].

Text Extraction plays a major role in finding vital and valuable information. Text extraction involves detection, localization, tracking, binarization, extraction, enhancement and recognition of the text from the given image. These text characters are difficult to be detected and recognized due to their

deviation of size, font, style, orientation, alignment, contrast, complex colored, textured background. Due to the rapid growth of available multimedia documents and growing requirement for information, identification, indexing and retrieval, many researches have been done on text extraction in images. Several techniques have been developed for extracting the text from an image. The proposed methods were based on morphological operators, wavelet transform, artificial neural network, skeletonization operation, edge detection algorithm, histogram technique etc. All these techniques have their benefits and restrictions. This article discusses various schemes proposed earlier for extracting the text from an image. This paper also provides the performance comparison of several existing methods proposed by researchers in extracting the text from an image [20].

2.2 Contribution

In this paper, a new method proposed to recognize a text based on features in this framework, consists of a formal model definition and three stages algorithm for recognition. Pre-processing image then Modified the classical way of Determinant value (that is requisite input whole the Handwritten text into determinant computation), so that the created recognition system becomes feasible (fast and accurate) for recognizing a query text from a huge dataset also, propose a suitable similarity measure incorporating the quantitative meaningful features (Det value) with the qualitative features to get a potential recognition decision.

3. THE RECOGNITION METHODOLOGY

3.1 Pre-processing

The pre-processing is a series of operations performed on the scanned input image. It essentially enhances the image; rendering it suitable for segmentation. The various tasks performed on the image in pre-processing stage are shown in Fig.1, Extract features. Binarization process converts a gray scale image into a binary image using global thresholding technique, dilating the image and filling the holes present in it are the operations performed in the last two stages to produce the pre-processed image suitable for Recognition [16].

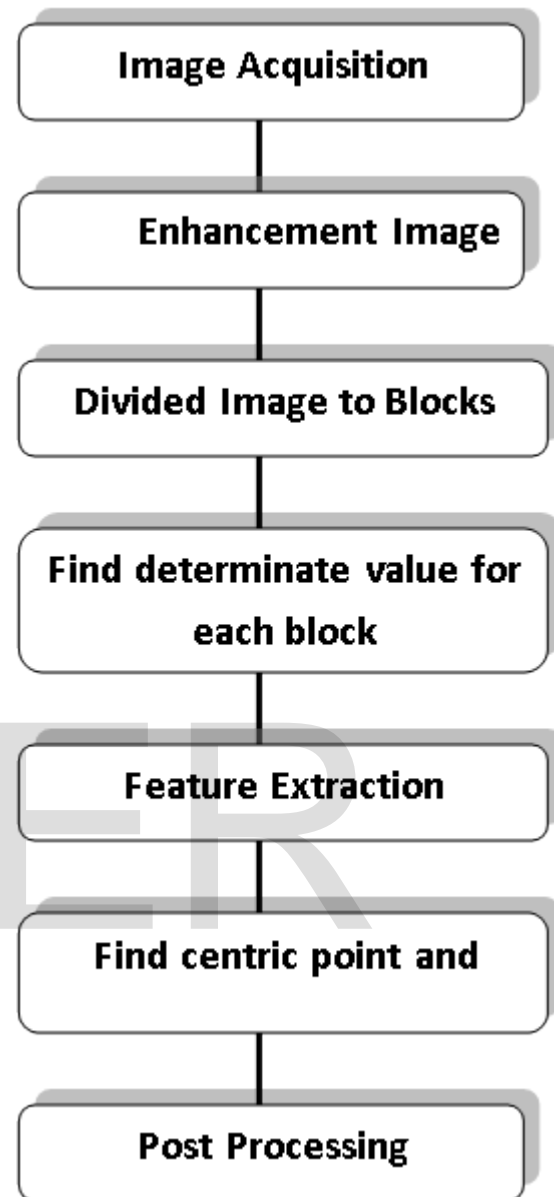


Fig.1 Image pre-processing Stage

3.2 Image Acquisition

In Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMP etc., as shown in Fig-2. This image is acquired through a scanner, digital camera or any other suitable digital input device.

- Khamael Abbas is currently Lecturer in computer science in Al-nahrainUniversity, Iraq, E-mail: khamail_abbas@yahoo.com
- Itimad Raheem is currently pursuing phd degree program computer science inUTM University, Malayazia, E-mail:wefee@yahoo.com



Fig-2 Scanned image of Handwritten Text

3.3 Enhancement Image

The median filter is normally used to remove the noise produced in an image; it often does a better job than the mean filter of preserving useful detail in the image. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value divided by 2. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values divided by 2 is used.) Fig-3 illustrates an example of this calculation.

| | | | | |
|-----|-----|-----|-----|-----|
| 123 | 125 | 126 | 130 | 140 |
| 122 | 124 | 126 | 127 | 135 |
| 118 | 120 | 150 | 125 | 134 |
| 119 | 115 | 119 | 123 | 133 |
| 111 | 116 | 110 | 120 | 130 |

Neighbourhood values:

115, 119, 120, 123, 124, 125, 126, 127, 150

Median value: 124 / 2

Fig-3 An example of the Calculation of Median Filter for Enhancement Image



Fig-4 Enhancement Image using Median Filter

3.4 Divided Image to Blocks

256x256 pixel image is divided into blocks of typically 3x3 pixels. For each block the Determinant values are calculated, these values change from block to block shown Fig-5.

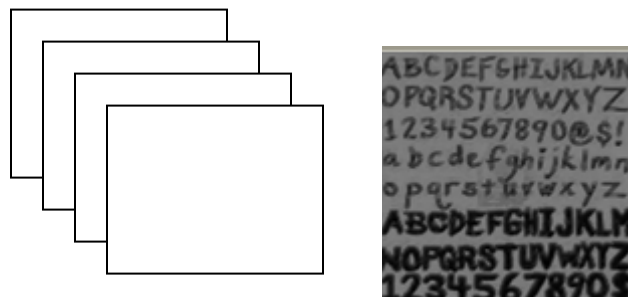


Fig-5 Divided Image to Blocks size (3x3)

3.5 Theory to Extract Feature using Determinant Value

The term determinant of a square matrix A, denoted $\det(A)$ or $|A|$, refers to both the collection of the elements of the square matrix, enclosed in vertical lines, and the scalar value represented by that array. Thus,

$$\det(A) = |A| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \dots \dots \dots \text{(Eq.1)}$$

Only square matrices have determinants. The scalar value of the determinant of a 2×2 matrix is the product of the elements on the major diagonal minus the product of the elements on the minor diagonal. Thus

$$\det(A) = |A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} a_{22} - a_{12} a_{21} \dots \dots \dots \text{(Eq.2)}$$

The scalar value of the determinant of a 3×3 matrix is composed of the sum of six triple products which can be obtained from the augmented determinant [6]:

$$\det(A) = |A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} \end{vmatrix} \dots \dots \dots \text{(Eq.3)}$$

The 3×3 determinant is augmented by repeating the first two columns, of the determinant on the right-hand side of the determinant. Three triple products are formed, starting with the elements of the first row multiplied by the two remaining elements on the downward-sloping diagonals. Three more triple products are formed, starting with the elements of the third row multiplied by the two remaining elements on the right-upward sloping diagonals. The value of the determinant is the sum of the first three triple products minus the sum of the last three triple products. Thus [6],

$$\det(A) = |A| = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12} \dots \dots \dots \text{(Eq.4)}$$

Threshold (T) was practically designed to check the value of image

$$\begin{aligned} &\text{If } im(I,j) > T \text{ im}(I,j) \\ &\text{Else } im(I,j) = 0 \end{aligned}$$



Fig.6 Extract Features using Determinate value for each block

3-6 Determine Centric Point using Chain Code Algorithm

An algorithm was developed to create the chain code for any thinned Features. Chain code representation of object contours is extensively used in document analysis and recognition. It has proven to be an efficient and effective representation, especially for handwritten documents. Unlike the thinned skeletons, chain code is a lossless representation in the sense that the pixel image can be fully recovered from its chain code representation. [19]. Chain codes are used to represent a boundary by connecting sequence of straight line segments of specified length and direction. Typically, this representation is based on the 4- or 8- connectivity of the segment where, direction of each segment is coded using an alphabet scheme such as; the one shown in Fig. (7). Since digital images are usually acquired and processed in a grid format with equal spacing in the x and y directions, the algorithm must choose which path to go first and must return back to the cross point to follow a different path. To solve this problem, the developed chain code is designed to classify the paths, either as a terminated path (as shown in Fig 7-a), forked path (as shown in Fig 7-b), or circular path (as shown in Fig- 7-c and Fig- 7-d). When the tracing reaches a cross point of many paths, it analyzes each path to know its type and the number of pixels it contains. Then, it sorts the traced paths; starting first with the circular paths, then the terminated paths and the forked paths at last.

The circular and terminated paths are sorted according to the number of pixels the path contains, forked paths are sorted according to the number of pixels the path contains to the cross point of other paths.

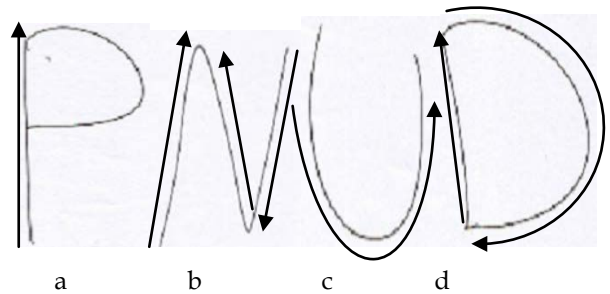


Fig.7 chain code algorithm and path type terminated path, forked path, circular path

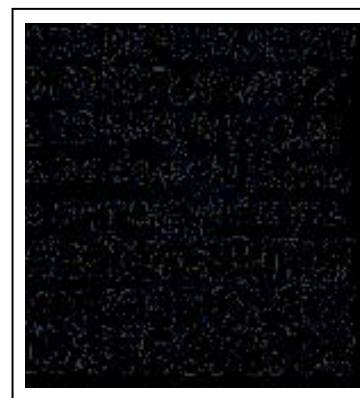
Then, the coordinates of all collected pixels are normalized to be invariant to translation and scaling. Eq. (5 - 7) were used for mapping x and y coordinates to their corresponding normalized values.

$$x_n = \frac{x - x_{\min}}{D} \dots \dots \dots \text{(Eq.5)}$$

$$y_n = \frac{y - y_{\min}}{D} \dots \dots \dots \text{(Eq.6)}$$

$$D = \max\{x_{\max} - x_{\min}, y_{\max} - y_{\min}\} \dots \dots \dots \text{(Eq.7)}$$

Where, n is the number of pixels of the traced text segment, x_n and y_n are the normalized values, x and y are the old pixel coordinates, x_{\min} is the smallest registered x coordinate value, x_{\max} is the largest registered x coordinate value, y_{\min} is the smallest registered y coordinate value, y_{\max} is the largest registered y coordinate value.



- H1 Center point for each block (10×10)
- X1,X2,X3 initial point for each block
- Background of image

Fig.8 Center point for each block using chain code

The Euclidean distance between each pixel (x, y) on the boundary and the centric point were designed, as eq. (8) shown in Table-1.

$$D(x, y) = \sqrt{(x - x_{centriod})^2 + (y - y_{centriod})^2}$$

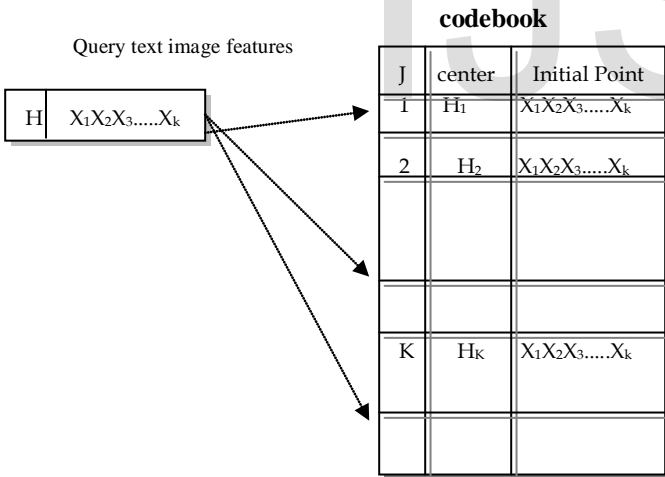
.....(Eq.8)

Using these key points, points X1, X2, X3, X4 were found to be the initial points and center points of the each block in the of image. A sample of the way to find these points, is shown in Fig.3. H1 (centric point) was found and then Euclidean distance of pixels on the boundary was calculated.

Finally, the recognition for partially occluded and overlapping objects in composite scenes is done. The objects are matched against the scene and their positions are recovered; not effectively applicable to ink of text, color of text and scale invariant cases. Rate of recognition is Successful [7]. As it is can be clearly seen, the similarity between the verified image text and the trained set can be represented by the minimum distance test (i.e. utilizing the Mean-Square-Error –MSE || criterion), given by:

$$Min\{MSE_K\} = Min\{\sum_{i=1}^M (V_{K,i} - V_{M+1,i})^2\}, \text{ for } K = 1, 2, \dots, M$$

.....(Eq.9)



4 EXPERIMENT AND EVALUATION

To test the general applicability of the proposed approach, the handwritten text recognition system was built to distinguish the query text in between dataset. The camera takes the image. This research shows that the shapes of pattern spectra differ from person to person. Features of text image are defined from Angular relation between characters' Region. In the following, an explanation about input/output details and the evaluation of the proposed system:

4.1 Input Handwritten Text Dataset

The handwritten text dataset are recorded by Scanner under windows operating system with an attribute of 24-BMP image. We used the database for fingerprints recognition experiments. Here, we have experimented with nearly 40 images with variations of 20 persons taken by 4 samples; male, female, and verify image is the same font of image-1 But, the character and scaling of font were different, as shown in Fig-(9-1).

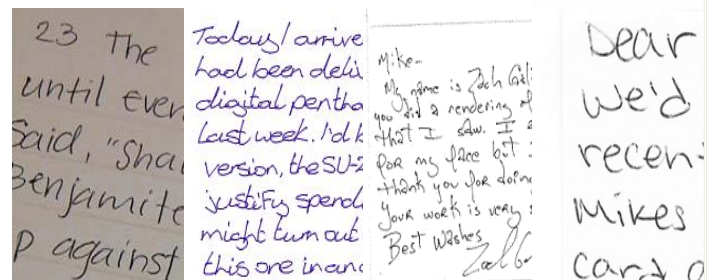


Fig. (9-1) Handwritten Text Dataset

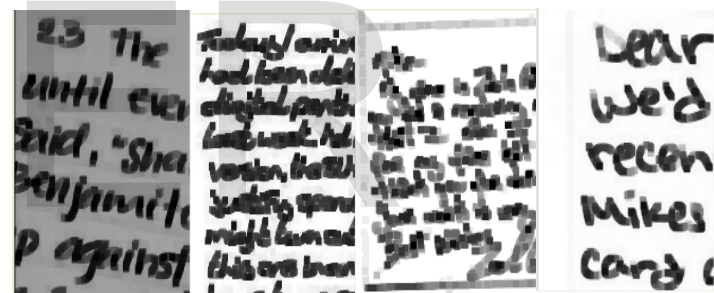


Fig. (9-2) Median Filter of Handwritten Text Dataset

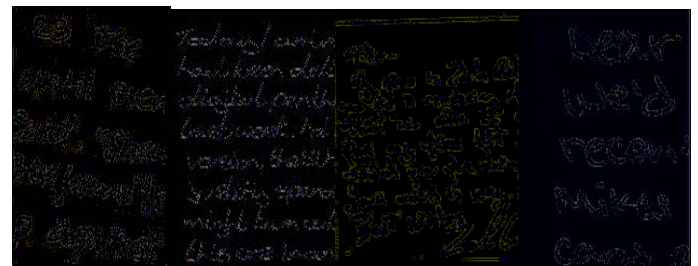


Fig. (9-3) Centric Point for Handwritten Text Dataset

4.2 Output Results

The experiment results show the effectiveness of the proposed method. They show clearly the flexibility of the method in relation to step translation of the handwritten Text. A preview image of the Database of text is as shown in Fig-(10). The experimental observations of the experiments performed on datasets are shown in Table-2 and Table-3 as follows:

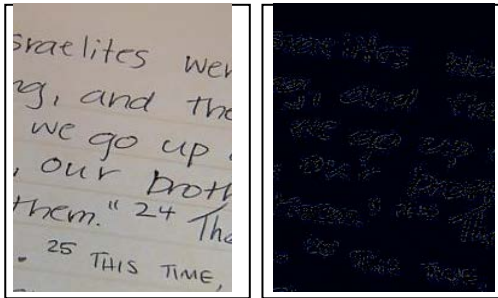


Fig10 a- Verify Image b-Extract Features to verify Image

The Euclidean Distance between trained set and the verifying handwritten text image is obvious, as listed below:

TABLE 1. EUCLIDIAN DISTANCE BETWEEN INITIAL POINT AND CENTRIC POINT FOR EACH IMAGE AND VERIFIED IMAGE

| Item | E D | E D | E D | E D | E D |
|------|--------|---------|---------|---------|---------|
| | Image1 | Image 2 | Image 3 | Image 4 | Image v |
| 1 | 0.203 | 12.309 | 3.415 | 4.521 | 0.115 |
| 2 | 9.834 | 11.742 | 13.65 | 15.558 | 3.11 |
| 3 | 2.6781 | 2.867 | 4.56 | 3.245 | 2.111 |
| 4 | 2.831 | 12.001 | 3.171 | 17.321 | 2.321 |
| 5 | 1.234 | 8.181 | 6.373 | 7.565 | 5.413 |
| 6 | 0.929 | 10.131 | 3.6671 | 4.465 | 2.323 |
| 7 | 4.0091 | 17.319 | 10.629 | 13.939 | 5.212 |
| 8 | 8.883 | 4.549 | 10.215 | 10.881 | 7.534 |

The Min {MSE} between trained set and the verifying Handwritten text image is obvious, between ED image1 and ED image vi.e. Min {MSEK}=MSE1, which is illustrated in Eq.(9) as listed below:

TABLE 2. MEAN SQUARE ERROR BETWEEN TRAINING SET OF TEXT IMAGE DATA SET AND VERIFY IMAGE

| NO | MSE1 | MSE2 | MSE3 | MSE4 |
|----|--------|--------|--------|--------|
| 1 | 0.088 | 12.194 | 3.3 | 4.406 |
| 2 | 6.724 | 8.632 | 10.54 | 12.448 |
| 3 | 0.5671 | 0.756 | 2.449 | 1.134 |
| 4 | 0.51 | 9.68 | 0.85 | 15 |
| 5 | 4.179 | 2.768 | 0.96 | 2.152 |
| 6 | 1.394 | 7.808 | 1.3441 | 2.142 |
| 7 | 1.2029 | 12.107 | 5.417 | 8.727 |
| 8 | 1.349 | 2.985 | 2.681 | 3.347 |

4.3 Evaluation

In the current computer implementation of determinant problem based class and mapping, a careful computation treatment was needed, especially with that related to the variables precision and the normalization of determinant values and using chain coding. Thereby, the numerical experiments indicate quite optimistic availability of the proposed algorithm for text recognition system. Suppose an existed image within the trained set has been selected to be verified, as illustrated in Fig (9-1), Fig (9-2), Fig (9-3) and Fig (10).

5. CONCLUSION

In this paper, a complete and fully automatic approach for Handwritten Text recognition images is proposed. The algorithms used in Handwritten Text recognition are categorized into three stages: gathering sample of handwritten text, pre-processing to bold text and remove noise, feature extraction and template matching. Our recognition makes the computationally approach inexpensive and significantly robust recognition decision using three-stage feature matching with centric point and the boundary edge of Handwritten Text measures, significantly improved the matching performance. Comparing with other methods a high recognition rate by verifying raw text directly is difficult to obtain, because people stand in various positions with different distances and angles between the characters and ink of pen and press on pen during writing. To achieve robustness in matching an input Handwritten Text, it must be normalized in position and direction. Such normalization might remove useful information for recognition

References

- [1] E. Augustin, M. Carr'e, G. E., J. M. Brodin, E. Geoffrois, and F. Preteux. Rimes evaluation campaign for handwritten mail processing. In Proceedings of the Workshop on Frontiers in Handwriting Recognition, pages 231-235, 2006.
- [2] R. Bertolami and H. Bunke. Hidden markov modelbased ensemble methods for offline handwritten text line recognition. 41(11):3452-3460, Nov. 2008.
- [3] H. Bunke. Recognition of cursive roman handwriting: past, present and future. In Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, pages 448 - 459 vol.1, aug. 2003.
- [4] R. G. Casey. Moment normalization of handprinted characters. IBM Journal of Research and Development, 14(5):548 -557, sep. 1970.
- [5] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney. Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: a comparison for offline handwriting recognition. In IEEE International Conference on Image Processing, Brussels, Belgium, Sept. 2011.
- [6] P. Dreuw, G. Heigold, and H. Ney. Confidence- andmargin-based mmi/mpe discriminative training for offline handwriting recognition. Int. J. Doc. Anal. Recognit., 14(3):273-288, Sept. 2011.
- [7] S. Espa'na-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid

- hmm/arn models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):767–779, april 2011.
- [8] A. Giménez, I. Khoury, and A. Juan. Windowed bernoulli mixture hmms for arabic handwritten word recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pages 533–538, nov. 2010.
- [9] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *31(5):855–868*, May 2009.
- [10] E. Grosicki and H. El Abed. Icdar 2009 handwriting recognition competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1398–1402, july 2009.
- [11] S. Jonas. Improved modeling in handwriting recognition. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, Jun 2009.
- [12] C.-L. Liu, M. Koga, H. Sako, and H. Fujisawa. Aspect ratio adaptive normalization for handwritten character recognition. In T. Tan, Y. Shi, and W. Gao, editors, *Advances in Multimodal Interfaces – ICMI 2000*, volume 1948 of *Lecture Notes in Computer Science*, pages 418–425. Springer Berlin / Heidelberg, 2000. 10.1007/3-540-40063-X 55.
- [13] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.
- [14] C.-L. Liu, H. Sako, and H. Fujisawa. Handwritten chinese character recognition: alternatives to nonlinear normalization. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 524–528 vol.1, aug. 2003.
- [15] U.-V. Marti and H. Bunke. The iam-database: an English sentence database for offline handwriting recognition. *5(1):39–46*, Nov. 2002.
- [16] T. Miyoshi, T. Nagasaki, and H. Shinjo. Character normalization methods using moments of gradient features and normalization cooperated feature extraction. In *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, pages 1–5, nov. 2009.
- [17] M. Pastor, A. Toselli, and E. Vidal. Projection profile based algorithm for slant removal. In A. Campilho and M. Kamel, editors, *Image Analysis and Recognition*, volume 3212 of *Lecture Notes in Computer Science*, pages 183–190. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-30126-4 23.
- [18] H. Pesch. Advancements in latin script recognition. Master's thesis, RWTH Aachen University, Aachen, Germany, Nov. 2011.
- [19] J.-F. Rivest, P. Soille, and S. Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336, 1993.
- [20] C.P. Sumathi¹, T. Santhanam² and G.Gayathri Devi³, || A SURVEY ON VARIOUS APPROACHES OF TEXT EXTRACTION IN IMAGES || *International Journal of Computer Science & Engineering Survey (IJCES)* Vol.3, No.4, August 2012 DOI : 10.5121/ijcses.2012.3403 27
- [21] Bremananth R and Prakash A, —Tamil Numerals Identification ||, *International Conference on Advances in Recent Technologies in Communication and Computing*, page(s): 620 – 622, 2009
- [22] Hewavitharana S and Fernando H.C, —A Two Stage Classification Approach to Tamil Handwritten Recognition ||, *Tamil In Internet, California, U SA*, 2002.
- [23] Indra Gandhi R and Iyakutti K, —An attempt to Recognize Handwritten Tamil Character using Kohonen SOM ||, *Int. J. of Advanced Networking and Applications*, Volume: 01 Issue: 03 Pages:188-192, 2009