RESEARCHERID
THOMSON REUTERS

[Mukund* *et al.*, 6(4): April, 2017]
ICTM Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ENGLISH CURSIVE SCRIPT RECOGNITION

Miss.Yewale Poonam Mukund*, Dr. M.S.Deshpande
* Electronics and Telecommunication, TSSM's Bhivarabai Sawant College of Engineering and Research, Pune, India
Electronics and Telecommunication, TSSM's Bhivarabai Sawant College of Engineering and Research, Pune, India

DOI: 10.5281/zenodo.556324

ABSTRACT

This system proposes a fast method for english cursive script recognition (ECSR) which presents three main contributions. The first one is Unwanted Lines Elimination Algorithm (ULEA) . After binarization the input image by using an adaptive thresholding, Unwanted Lines Elimination Algorithm (ULEA) is proposed to enhance the image. The second contribution is that our proposed ECSR method processes very low-resolution images taken by a scanner. After the vertical edges have been detected by ULEA, the most-like character details based on feature information are highlighted. Then, the sample region based on statistical and logical operations will be extracted. The third contribution is character classification is achieved by using support vector machines (SVMs). A database of 1080 characters was used to train and test the cursive character recognizer. SVMs compare notably better, in terms of recognition rates, with popular classifiers, SVM recognition rate is among the highest presented in the literature for cursive character recognition.

KEYWORDS: Off-line cursive handwriting recognition, optical handwritten character recognition, preprocessing, feature extraction, Support vector machines

INTRODUCTION

Offline handwriting recognition is the task of determining what letters or words are present in a digital image of handwritten text. It is of significant benefit to man-machine communication and can assist in the automatic processing of handwritten documents. It is a subtask of Optical Character Recognition (OCR), whose domain can be machine-print or handwriting but is more commonly machine-print. The recognition of cursive handwriting presents unique challenges and benefits and has been approached more recently than the recognition of text in other scripts. This paper describes the state of the art of this field. A recognition system can be either “online” or “offline.” It is “online” if the temporal sequence of points traced out by the pen is available, such as with electronic personal data assistants that require the user to “write” on the screen using a stylus. It is “offline” if it is applied to previously written text, such as any images scanned in by a scanner. The online problem is usually easier than the offline problem since more information is available. is survey is restricted to offline handwriting systems.. independent transcription structure that could be used to copy incoming mail, or faxes, address blocks, or checks. This paper is organized as follows. Section II introduces a brief of related work. Section III and section IV describes pre-processing and feature extraction respectively.

Section V includes classification method and Experimental results and discussion are presented in Section VI. Section VII draws conclusion.

LITERATURE SURVEY

Recognizing correctness of the image depends on the compassion of the designated features and type of classifier used. Hence, number of feature extraction and classification approaches are proposed by different reserchers. Following paper achieve handwritten character recognition of cursive words. Radmilo M. Bozinovic and Sargur N. Srihari [1] used the rounded way for cursive word recognition. The method used here is to signify word through several stages of conversion like points, contours, features, letter and word. A unique feature vector is produced from the image using statistical requirements among letter and feature; partially calculated words are recognized

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

by associating with lexicon. Lexicon includes 130 words, thus limited no of words are recognized. Classifiers are not used for recognization of words, rating is given to each segment which are parted by pre-segmentation using letter premise and they are recognized based on supreme value of evaluation H. Bunke, M. Roth and E. G. Schukat-Talamazzini [2]. They extract features from the skeleton of word. The author had completed recognition both cursive and inaccessible handwritten characters using HMM. Hybrid way is used to maximize the power of HMM. For recognition of characters features used are medians of black run in every one scan line. Character image is perused in four dissimilar orders for extracting feature. Medians in each track represent a sparse indicator skeleton of the character. The discrete density left to right HMM is used for recognition. Yong Haw Tay, Pierre-Michel Lallican, Marzuki Khalid, Christian Stefan Knerr (2001) [4]. The researcher presented handwritten cursive words using recognition based segmentation process and it gives the judgment between two systems. The first recognition system uses mixture of Neural Network and HMM aimed at recognition. In second process discrete HMM is castoff. It first process Pre-segmentation of word is done by segmentation chart. Neural network analyses the possibility for each letter premise in chart and then HMM computes possibility for each word in wordlist by adding the chance along each possible path in grid. In second manner 140 ordered features are take out from each segment which is parted by pre segmentation. This features through vector quantization transformed to only symbol and lastly by calculating the chance for each word in lexis word is recognized. Anshul Gupta, Manisha Srivastava (2011) [5] .In this paper author castoff segmentation grounded method for cursive word credit. In this system cursive words are segmented into discrete characters, which are than recognized and fused to harvest meaningful word by likening with lexicon. They have considered only 26 words. Use of support vector machines for character recognition is found in [02]. *A. De Gasperi* [02] implemented a support vector machines approach as classifier SVMs compare notably better, in terms of recognition rates.

Still cursive script recognition is challenging problem, here we have proposed CSR by using unwanted lines elimination segmentation algorithm and support vector machine as classifier.

PROPOSED RECOGNITION SYSTEM

This paper has three contributions pre-processing, extraction and recognition. The proposed CSR system is shown in Fig. 2. Different phases of CSR are discussed below.

PRE-PROCESSING

Imag Acquisition:

Fig.(1) Input Image

The images are acquired from flatbed scanner and a training database of 1080 cursive characters is prepared .The purpose of a recognition system is to transcribe "*words*" into "words".

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

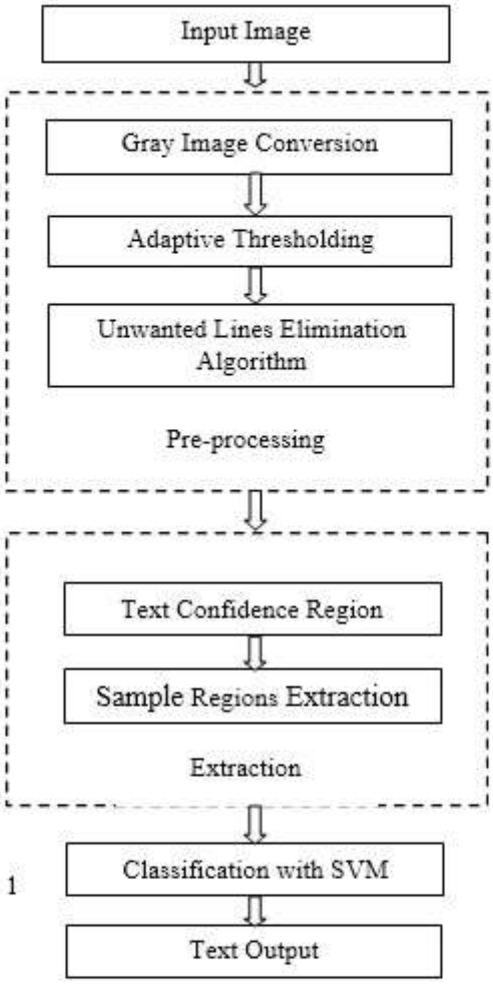

Fig.(2) System block diagram

Gray Image Conversion and Adaptive Thresholding:

The input image is converted to gray scale image then an adaptive thresholding process [12, 13] is applied in order to constitute the binarized image. In order to get a good adaptive threshold, the method proposed in [13] is used. Wellner's algorithm depends on the scanning order of pixels. Since the neighborhood samples are not evenly distributed in all directions, the moving average process is not suitable to give a good representation for the neighboring pixels. Therefore, using the integral image in [12] has solved this problem.

Fig.(3) Thresholding Image

Adaptive Thresholding Formulations:

The first step is to compute the integral image. Initially, the summation of the pixel values for every column *jth* through all row values *i*'s will be computed as follows:

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

$$sum(i) = \sum_{x=0}^{i} g(x,y) \quad (1)$$

where $g(x,y)$ represents the input values and $sum(i)$ represents all cumulative gray values of $g(x,y)$ for the column *jth* through all rows of image, *i=0, 1, ... height*.
Then, the integral image can then be computed for every pixel as follows:

If $\&j = 0$
$$interal\ image(i,j) = sum(i) \quad (2)$$
Otherwise,
$$interal\ image(i,j) = integral\ image(i,j-1) + sum(i),$$

The next step is to perform thresholding for each pixel. In order to do so, first, the intensity summation for each local window should be computed by using two subtraction and one addition operations by using above equation. where sum_{window} represents the summation of the intensities of the gray values for a specified local window in which the currently binarized pixel is centering in. The boundaries of window can be represented by:

$$(i+\tfrac{s}{2}, j+\tfrac{s}{2}),(i+\tfrac{s}{2}, j-\tfrac{s}{2}),(i-\tfrac{s}{2}, j+\tfrac{s}{2}),(i-\tfrac{s}{2}, j-\tfrac{s}{2})$$

And s represents the local window size/ lengths for the computed integral image whereas
$$s=\frac{Image\ width}{8} \quad (3)$$
Therefore, in order to compute the adaptive threshold value for the image in which $g(i,j)$ € [0,255] is the intensity of
Pixel located at (i,j), the threshold $t(i,j)$ for each pixel has to be computed first as in eq (4).
$$t(i,j) = (1-T) * sum_{window} \quad (4)$$
where $t(i,j)$ represents the threshold for each pixel at (i,j) location and T is a constant; *T=0.15*.)

Unwanted Lines Elimination Algorithm (ULEA)
Thresholding procedure in general produces many reedy lines which do not belong to text region. These background and noise edges are unwanted lines. These lines may interfere in the text location. Therefore, we have proposed an algorithm to eliminate them from the image. ULEA has been proposed in order to eliminate these lines. In this step, while processing a binary image, the black pixel values are background, and the white pixel values are foreground. A 3×3 mask is used throughout all image pixels. Only black pixel values in the thresholded image are tested. In order to retain small details of license plate, only the lines whose widths equal to 1-pixel are checked. Supposed that *b(x,y)* are the values for thresholded image. Once, the current pixel value located at the mask center is black, the 8-neighbor pixel values are tested. If two corresponding values are white together, then the current pixel is converted to white value as foreground pixel value (i.e., white pixel).

Fig.(4) Unwanted Lines

Fig. 4 shows the output after ULEA is performed, whereby many unwanted lines are disappeared from the image. This kind of image is nearly ready for better segmentation process.

RESEARCHERID
THOMSON REUTERS

[Mukund* *et al.*, 6(4): April, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

FEATURE EXTRACTION

Text Confidence Region Based on ULEA (TCR):

After applying ULEA, the next step is to highlight the desired details such as character details and vertical edges in the image. TCR performs NAND-AND operation for each two corresponding pixel values taken from ULEA output images.

This process depends on ULEA output in highlighting the text region. All the pixels in vertical edge image will be scanned. When there are two neighbor black pixels and followed by one black pixel as in ULEA output form, the two edges will be checked in order to highlight the details by drawing white horizontal lines connecting each two vertical edges. First, these two vertical edges should be surrounded by a black background as in the ULEA image Second; the value of horizontal distance (hd) represents the length between the two vertical edges of a single object. The hd has been computed using the test images. The *hd* value is selected to be suitable for removing long foreground and random noise edges that have not been eliminated earlier. This scanning process will start moving from left to right and from top to bottom.

Sample Regions Extraction (SRE)

This process is divided into four steps as follows:

1) Count the Drawn Lines per Each Row: The number of lines that have been drawn per each row will be counted and stored in a matrix variable, *how many line*[*a*], where *a= 0, 1, … height-1*.

2) Divide the Image into Multi-groups: The huge number of rows will delay the processing time in next steps. Thus, in order to reduce the consumed time, gathering many rows as a group is used here. Therefore, dividing the image into multi-groups could be done as follows

$$\text{How\_ many\_ groups} = \frac{Height}{c} \qquad (5)$$

Where above equation represents the total number of groups, *height* represents the total number of image rows, and *c* represents SRE constant. In our work, *c* is chosen to represent one group (set of rows). For our methodology, *c*=10, because each 10 rows could save the computation time Also, it could avoid either loosing much details or consuming more computation time for processing the image.

Therefore, each group consists of ten rows. Due to the *How_many_Lines* values, some rows have number of drawn horizontal lines, and this makes some groups have horizontal lines. The step here is to store the total number of horizontal lines for each group. In this step, a matrix is created to store the total number of drawn lines for each ten rows (group).

Fig.(5) Unwanted Lines Elimination

3) Count and Store Satisfied Group Indices and Boundaries: Most of the group-lines are not parts of the word details. So, it is useful to use a threshold in order to eliminate those unsatisfied groups and keep the satisfied ones in which the character details exist in. Each group will be checked; if it has at least 15 lines then it is considered as a part of the character region. Thus, the total number of groups including the parts of character regions will be counted and stored. The remaining groups after thresholding step should have the character details. Therefore, their locations are stored. The final step here is to extract both upper and lower boundaries of each satisfied group by using its own index.

This threshold value is determined in order to make sure that the small-sized character is included for recognition process. If the pre-defined threshold is selected with less than this value, wrong result can be yielded because noise and / or non-text regions will be considered as parts of the true text whereas they are not. Based on that, the optimal threshold for the best detection rate has been found is $\geq \frac{1}{20} \times$ image height

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Therefore threshold in our case is $\geq \frac{1}{20} \times 288 \geq 14.4 = 15$

4) Select Boundaries of Candidate Regions: This step is to draw the horizontal boundaries above and below each sample region. Fig. 5 shows the result of drawing sample regions boundaries in the input image. As can be seen, there are two candidate regions interpreted from horizontal lines plotting and these conditions require additional step before the text region can be correctly extracted.

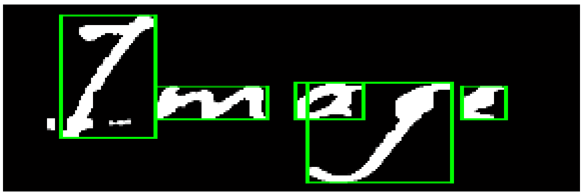

Fig.(6) Boundries of Candidate Region

CLASSIFICATION

In order to use SVM when the number of classes *K* is larger than 2, a few different strategies have been suggested [13]. In our experiments we have adopted *one versus others* (*o-v-o*) method [14,15]. The method learns one classifier for each of the *K* classes against all the other classes. More formally, the method consists in training *K* SVM classifiers *fj* by labeling all training points having $y_i = j$ with +1 and $y_i \neq j$ with −1 during the training of the *j* th classifier. In the test stage, the final decision function *F(·)* is given by

$$F(\vec{x}_i) = \arg\ max f_j\ (\vec{x}) \quad (7)$$

EXPERIMENTAL RESULTS AND DISCUSSION

A database of 1080 characters is used to train the cursive character recognizer. and 70 cusrive words are used for testing. in which 30 words are three letter words, 20 words are four letter words and 20 words are five letter words. we have tested total 30 samples of three different words, "set", "for" and "man". table 1 represents recognition accuracy of proposed method.

The correct segmentation does not always mean the extraction of a single character. It may divide a character like u and w into more than one segment, but it is not supposed to leave two distinct characters together. It is observed that correct segmentation depends mostly on the determination of the segmentation regions.

The performance of proposed recognition method for 3letters, 4 letters and 5 letters words, which is shown in following table II.

Table I Recognition accuracy of three letter words

Accuracy	Set	For	Man
Proposed method	92%	90%	91%

Table II Recognition results of proposed system

No.of letters	3 letters	4 letters	5 letters
Recognition accuracy	90%	80%	85%

CONCLUSION

Today, machine-printed text documents with simple layouts can be recognized reliably by off-the-shelf OCR software. As we have seen throughout this paper, there is also some success with handwriting recognition, particularly for cursive words. Most of the off-line successes have come in constrained domains, such as postal addresses [16], bank checks, and census forms. In this we have implanted one new system which is combination of segmentation and recognition. For segmentation there is ULEA algorithm used to segment characters from cursive word. SVM used as classifier for better results. Therefore our results are far better than existing methods.

REFERENCES

[1] S. N. Huda, *et al.*, "Comparison of feature extractors in license plate recognition," in *Proceedings of the 1st IEvesEE Asia International Conference on Modelling & Simulation (AMS'07)*, Thailand, 2007, pp. 502-506.

[2] R. Plamondon, S. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. (2000) 63–84.

[3] A. Senior, A. Robinson, An off-line cursive handwriting recognition system, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 309–321.

[4] H. Bai and C. Liu, "A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology," in *Proceedings of the 17th International Conference on Pattern Recognition*, UK, 2004, pp. 831-834

[5] M. Fukumi, *et al.*, "Neural Network Based Threshold Determination for Malaysia License Plate Character Recognition," in *Proceedings of the 9th International Conference on Mechatronics Technology*, Malaysia, 2005, pp. 1-5.

[6] E. R. Lee, *et al.*, "Automatic Recognition of a Car License Plate Using Color Image Processing," in *Proceedings of the IEEE International Conference on Image Processing*, USA, 1994, pp. 301-305.

[7] R. Parisi, *et al.*, "Car Plate Recognition by Neural Networks and Image Processing," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, USA, 1998, pp. 195-198.

[8] T. Naito, *et al.*, "Robust License-Plate Recognition Method for Passing Vehicles under Outside Environment," *IEEE Trans. on Veh. Technol.*, vol. 49, pp. 2309-2319, 2000.

[9] H.-H. P. Wu, *et al.*, "License Plate Extraction in Low Resolution Video," in *Proceedings of the IEEE 18th International Conference on Pattern Recognition*, Hong Kong, 2006, pp. 824-827

[10] S. N. Huda, *et al.*, "Comparison of feature extractors in license plate recognition," in *Proceedings of the 1st IEEE Asia International Conference on Modelling & Simulation (AMS'07)*, Thailand, 2007, pp. 502-506

[11] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," Journal of Graphics Tools, vol. 12, pp. 13–21, 2007.

[12] F. Shafait, et al., "Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images," in Document Recognition,2007.