Handwritten English Character Recognition Using Edge Detection, Segmentation and Pattern Matching

Asoke Nath[1] , Srijita Kabasi[2] , Kasturi Chakraborty[3] , Sriya Sarkar[4]
[1]Associate Professor, Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[2]M.Sc. Student, Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[3]M.Sc. Student, Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[4]M.Sc. Student, Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
Corresponding Author: Asoke Naths

Abstract*: To convert handwritten character to printed character is a very challenging problem in computer science and information technology. It can save time and makes human's job easier. Handwritten Character recognition is an area of pattern recognition that has become the subject of research during the last some decades and has attracted many researchers across the world. Many researchers have proposed different approaches for character recognition in different languages. In this paper, a system is developed for the recognition of basic characters (vowels, consonants and numbers) in handwritten and printed English text, which can handle different font sizes and font types. The main important phases of character recognition include pre-processing, segmentation, feature extraction and matching.*

Keywords*: Basic Characters; English text; Handwritten character; Offline Character recognition; Online character recognition.*

Date of Submission: 07-06-2018 Date of acceptance: 22-06-2018

I. Introduction

Handwriting recognition is in research for over four decades and has attracted many researchers across the world. Character recognition systems translate such scanned images of printed, typewritten or handwritten documents into machine encoded text. This translated machine encoded text can be easily edited, searched and can be processed in many other ways according to requirements. Character recognition systems help humans ease and reduce their jobs of manually handling and processing of documents. Computerized processing to recognize individual character is required to convert scanned document into machine encoded form. Character recognition is mainly of two types online and offline. In online character recognition, data is captured during the writing process with the help of a special pen on electronic surface. In offline recognition, prewritten data generally written on a sheet of paper is scanned.

1) **Offline Character Recognition:** Generally all printed or type-written characters are classified in offline mode. Off-line handwritten character recognition refers to the process of recognizing characters in a document that have been scanned from a surface such as a sheet of paper and are stored digitally in grey scale format.

2) **Online Character Recognition:** The online mode of recognition is mostly used to recognize only handwritten characters. In this the handwriting is captured and stored in digital form via different means. Usually, a special pen is used in conjunction with an electronic surface. As the pen moves across the surface, the two- dimensional coordinates of successive points are represented as a function of time and are stored in order. This by handwriting on a pad instead of by typing using keyboard.

We have been successful in building the offline character recognition model and implementing the online character recognition model remains a future scope as of now. Also another limitation of this system is that it cannot correctly extract and recognize cursive handwriting.

II. Methods Used In Recognizing Handwritten Characters

The approach used in this paper is summarized as below:

It first takes the input image from user. It reads the image by an inbuilt function and then takes the whole image in a 2d array form where each element is the value of each pixel. Then it performs rgb2binary operation to convert all the pixel value to 1 or 0. That means it takes the drawn image pixels as value 1 and other white pixels as 0. After binarization the edges of the image are detected by selecting the top, left , right and bottom edge of the image by finding the last column having a black pixel. The first character is then segmented and is kept in a 2d matrix.

Then the first character is used for match operation. Before the matching it first resizes both the input character image and database image in same resolution otherwise the match operation can produce erroneous results. The resize operation is done using an inbuilt function “IMRESIZE” of MATLAB.

IMRESIZE(A, [NUMROWS NUMCOLS],’nearest’) resizes the image so that it has the specified number of rows and columns. Where A is the input image and NUMROWS and NUMCOLS is the number of rows and columns in the resized image. Either NUMROWS or NUMCOLS may be NaN, in which case IMRESIZE computes the number of rows or columns automatically in order to preserve the image aspect ratio. 'nearest' is used for nearest-neighbour interpolation.

The following operations are performed on the input image:

Image Acquisition: First we scan a handwritten character image and we apply the following procedures.

Image Pre-Processing : This consists of the processes Binarization and Edge detection.

In **binarization** it takes the original scanned image matrix as input and processes each pixel of it in such a way that, if the pixel value is greater than 100 it assumes it as a white pixel and change it to value 1. Similarly if the pixel value is less than 100 it assumes it as a black pixel and change it to 0.

In **filtering** a 3X3 window is taken and if there is atleast 5 white pixels then the pixel at the center of the window is considered to be noise and is set to white.

In **Edge detection** the boundary of input image is detected. It takes the binary image matrix as input, scan it from top row to bottom row for every pixel of each row. If it founds any pixel in a row then it assumes it as the top of image. After that it again scans all pixel of every row and if it founds any pixel then it will increment the row number. At the end of image, it will save the last row where it last found a black pixel in the image, and assume that row as the bottom of the image. Similarly by the same process it will detect the right and left edge of that image. It will scan it from left most column to right most column for every pixel of each column. If it finds any pixel in a column then it assumes it as the left most edge of image. After that it again scans all pixel of every column and if it finds any pixel it will increment the column number. At the right most end of image it will save the last column number where it found the last black pixel in the image, and assume that column as the right most edge of the image.

Crop Image: The left edge , right edge , top edge and bottom edge of the entire image are detected and cropped.

Character Segmentation: It takes the binary image as input and scan the columns until a column is found which contains all white pixel (0).This column with all 0 indicates a space. The system start scanning the segmented image column wise, as soon as it gets a black pixel in a column, it returns the column index as the left boundary edge of the character. The scan continues and when the system finds a column with all 0s, it returns the pervious left column index as the right boundary edge of the character. The resultant image is stored in a separate file by eliminating the portion on the left side of the left boundary edge and the portion on the right side of the right boundary edge.

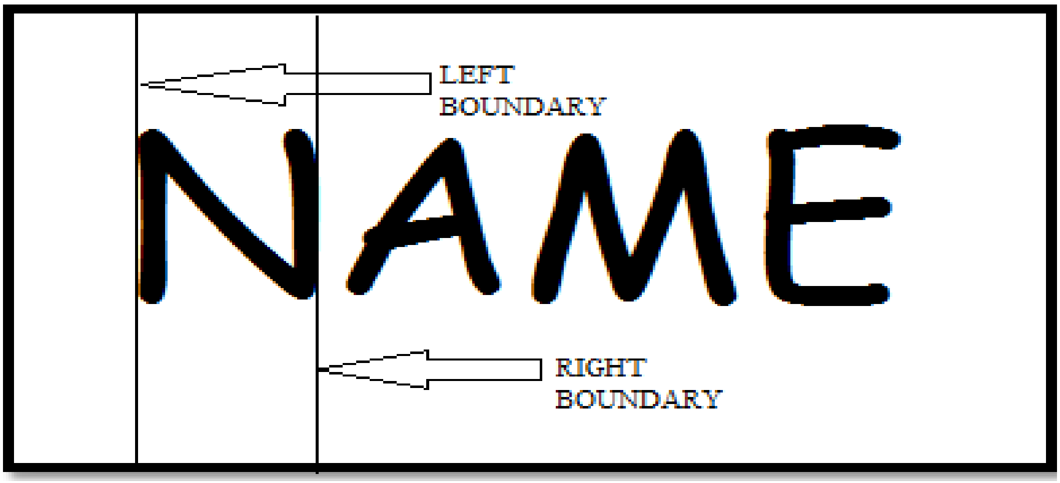


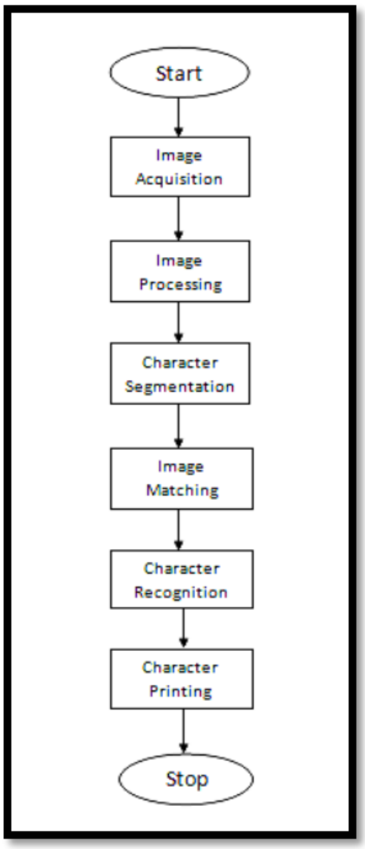
Fig 1: Character boundary detection

Character Image Matching: In this process, the matching operation of an input image with the image that is stored in the dataset is performed. Before matching, each image from the dataset is taken and binarized. Then both the acquired input character image and the binarized dataset image is resized in a particular size. Then it performs pixel by pixel matching between the input image and all the stored image in the dataset. It does the same for all the dataset image and the one with which the maximum number of pixels matched is selected and accordingly the appropriate character is detected from the data array.

We had preliminary proceeded with this approach. But we were not getting satisfactory result in terms of accuracy. In order to increase the accuracy, we are now using matching by correlation approach. In this method, the correlation co-efficient (which typically ranges between -1 and 1) between the acquired character image and every image from the dataset is calculated. The correlation co-efficient is calculated based on Pearson's method of correlation coefficient.

Character printing: Along with the dataset of images there is also a data array of characters that is maintained in the exact order as of the dataset images. The character with the maximum value of the correlation coefficient is recognized as the identified character and its respective index from the data array is taken which is then printed on the screen.

The following Block diagram gives the stepwise procedure for detecting characters:



III. Proposed Algorithm

Binarization:

```
Step 1: Start
Step 2: Convert RGB image P to gray scale image PGray using inbuilt function.
Step 3: Calculate size of PGray. 'r' denotes the number of rows and 'c' denotes  the number of  columns.
Step 4: Set i := 1
Step 5: Set j := 1
Step 6: If PGray(i,j) >= 100, set B(i,j) = 1
Step 7: Otherwise, set B(i,j) = 0
Step 8: Set j := j+1
Step 9: If j < c, go to step 6
Step 10: Set i := i+1
Step11: If i < r, go to step 5
Step 12: Stop
```

Filtering:

```
Step 1: Start
Step 2: Calculate size of B. 'r' denotes the number of rows and 'c' denotes the  number of columns
Step 3: Set the pixels at the first row, last row, first column and last column to 1
Step 4: Set i := 2
Step 5: Set j := 2
Step 6: Set count := Sum of the pixels in the range B(i-1,j-1) to B(i+1,j+1)
Step 7: If count > 5, set B(i,j) := 1
Step 8: Set j := j+1
Step 9: If j < c-1, Go to step 6
Step 10: Set i := i+1
Step 11: If i < r-1, go to step 5
Step 12: Stop.
```

Edge Detection:

```
Step 1: Start
Step 2: Calculate size of B. ‘r’ denotes the number of rows and ‘c’ denotes the number of columns
Step 3: Set n := 0
Step 4: Set i:= 1
Step 5: Set j := 1
Step 6: If B(i,j) = 0, set rt(rowtop) = i, set n = 1 and go to step 9
Step 7: Set j := j+1
Step 8: If j < c, go to step 6
Step 9: If n = 1, go to step 12
Step 10: Set i := i+1
Step 11: If i < r, go to step 5
Step 12: Set i := rt
Step 13: Set j := 1
Step 14: If B(i,j) = 0, set rb(rowbottom) = i
Step 15: Set j := j+1
Step 16: If j < c, go to step 14
Step 17: Set i := i+1
Step 18: If i < r, go to Step 13
Step 19: Set n := 0
Step 20: Set i:= 1
Step 21: Set j := 1
Step 22: If B(i,j) = 0, set cl(columnleft) = i, set n = 1 and go to step 25
Step 23: Set j := j+1
Step 24: If j < r, go to step 22
Step 25: If n = 1, go to step 28
Step 26: Set i := i+1
Step 27: If i < c, go to step 21
Step 28: Set i := cl
Step 29: Set j := 1
Step 30: If B(i,j) = 0, set cr(columnright) = i
Step 31: Set j := j+1
Step 32: If j < c, go to step 30
Step 33: Set i := i+1
Step 34: If i < r, go to Step 29
Step 35: Stop
```

Crop Image:

```
Step 1: Start
Step 2: Set k := 1
Step 3: Set l := 1
Step 4: Set i := top edge of the image(i.e. rt from edge detection algorithm)
Step 5: Set j := left edge of the image(i.e. cl from edge detection algorithm)
Step 6: If B(i,j) = 0, set S(k,l) := 0
Step 7: Otherwise, set S(k,l) := 1
Step 8: Set l := l+1
Step 9: Set j := j+1
Step 10: If j < right edge of the image(i.e. cr from edge detection algorithm), go to step 6
Step 11: Set S(k,l) := 1, k := k+1 and l := 1
Step 12: Set i := i+1
Step 13: If i < bottom edge of the image(i.e. rb from edge detection algorithm), go to step 5
Step 14: Stop
```

Character Edge Detection:

```
Step 1: Start
Step 2: Calculate size of S. 'r' denotes the number of rows and 'c' denotes the  number of columns
Step 3: Set x := 0
Step 4: Set i := left edge of the character( i.e. sc from detect space algorithm, initially sc = 0)
Step 5: Set j := 1 and m := 0
Step 6: If S(i,j)= 0, set m := 1 and go to step 10
Step 7: Otherwise, set m := 2
Step 8: Set j := j+1
Step 9: If j<r got o step 6
Step 10: If m = 2, set x=i-1 and go to step 13
Step 11: Set i := i+1
Step 12: If i<c go to step 5
Step 13: Stop
```

Character Segmentation:

```
Step 1: Start
Step 2: Calculate size of S. 'r' denotes the number of rows and 'c' denotes the number of  columns
Step 3: Set k := 1 and l := 1
Step 4: Set i := 1
Step 5: Set j := left edge of the character( i.e. sc from detect space algorithm, initially sc = 0)
Step 6: If S(i,j) = 1, set A(k,l) = 1
Step 7: Otherwise, set  A(k.l) = 0
Step 8: Set l := l+1
Step 9: Set j := j+1
Step 10: If j < right edge of the character(i.e. x from character edge detection algorithm) go  to step 6
Step 11: Set k := k+1 and l := 1
Step 12: Set i := i+1
Step 13: If  i < r, go to step 5
Step 14: Stop
```

Space Detection:

```
Step 1: Start
Step 2: Calculate size of S. 'r' denotes the number of rows and 'c' denotes the  number of columns
Step 3: Set m := 0
Step 4: Set j := x+1(x is the right edge of the character)
Step 5: Set i := 1
Step 6: If S(i,j) = 0, set m := 1 and go to step 10
Step 7: Otherwise, set m := 2
Step 8: Set i := i+1
Step 9: If i < r , got o step 6
Step 10: If m = 1, set sc := j and go to step 13
Step 11: Set j := j+1
Step 12: If j < c , got o step 5
Step 13: If m = 0, set sc := c
Step 14: Stop
```

Character Matching:

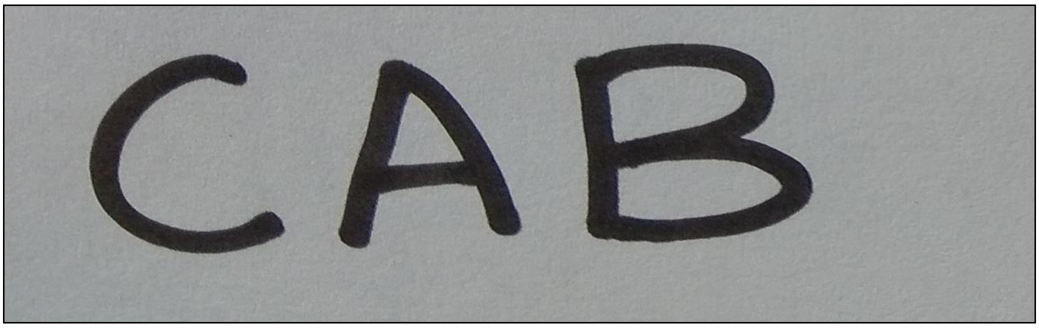
```
Step 1: Start.
Step 2: Set A to be the image to be matched //A is the input image
Step 3: for I : = each image of the data set
Step 4: if match (B ,I)= true , I is the image which is the best match of B
Step 5: Display Data(I), Data contains the machine encoded characters in the same order as the   datasets
Step 6: Stop
```

match(A,B)– this function accepts two images and matches them according to correlation coefficient matching technique.

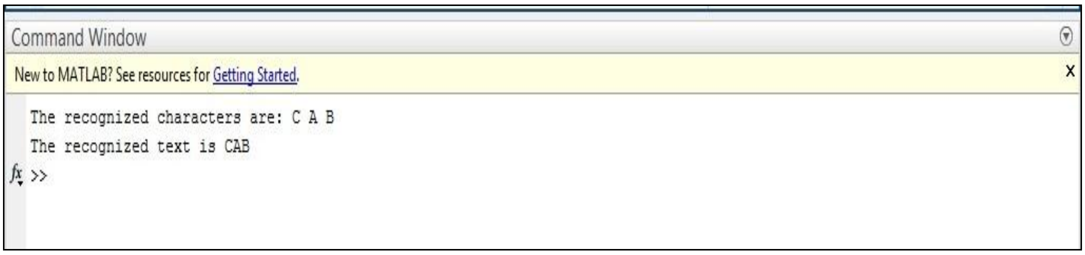
IV. Result And Discussion

Set 1:

Input:



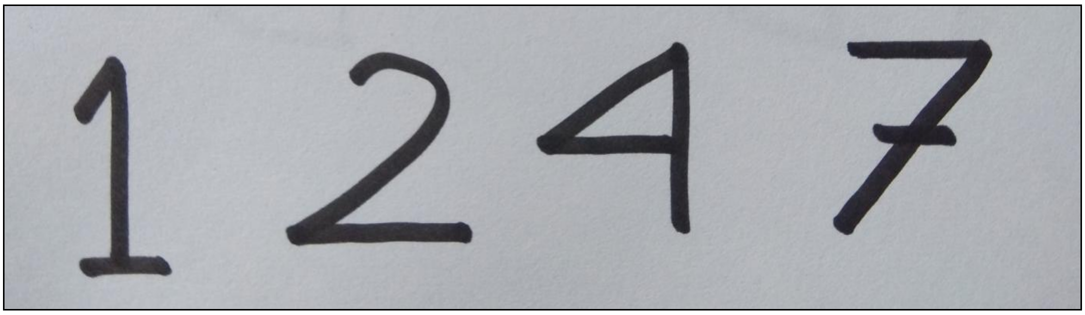
Output:



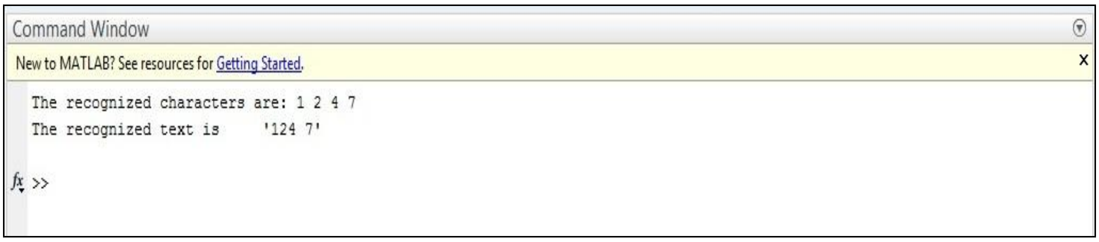
```
The recognized characters are: C A B
The recognized text is CAB
>>
```

Set 2:

Input:



Output:



```
The recognized characters are: 1 2 4 7
The recognized text is    '124 7'

>>
```

V. Limitations Of The System

This phase focuses on the two separate aspects: one which encountered during testing and another which we could not modify within the given time.

The 1st aspect includes-

(i) Our system is not capable of recognizing poor handwriting.

(ii) It can identify different font sizes but not different font types.

(iii) There should be a space between each consecutive characters because we are extracting the characters considering a space.

The 2nd aspect includes-
(i) The system cannot be applied to recognize a cursive handwriting Recognition.

VI. Conclusion And Future Scope

In the present paper handwritten character recognition is categorized into three main divisions: Image Acquisition, Image processing and Matching with the dataset. The experimental results illustrate how an input image leads us to character extraction and recognizes the handwritten characters accordingly. The designed systems have the ability to yield accurate results, provided the correct dataset is available. There is a scope of improvement in the current system using Neural Network. Hence, a simple yet effective approach for recognition of handwritten characters has been described. The recognition accuracy of the prototype implementation is promising, but more work needs to be done. In particular, no fine-tuning of the system has been done so far. Our character segmentation method also needs to be improved so that it can handle a larger variety of touching characters, which occur fairly often in images obtained from inferior-quality printed material. In general, the system needs to be tested on a wider variety of images containing characters in diverse fonts and sizes. This will enable us to identify the major weaknesses in the system and implement remedies for them. This program may detect small letter characters and special characters with the proper creation of datasets.

References

[1]. Monish Dutta[1], Pritha Roy[2], Sunanda Datta[3], Asoke Nath[4], “A New Method for Recognition of Handwritten Characters Using Edge Detection, Segmentation and Pattern Matching”, International Journal of Innovative Research in Computer and Communication Engineering, (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 6, June 2015, [1,2,3,4]Department of Computer Science, St. Xavier’s College (Autonomous), Kolkata, India.

[2]. Anshul Gupta and Manisha Srivastava , “offline handwritten character recognition” , Department of electronics and communication engineering, Indian Institute of Technology, Guwahati, Assam, India – 781039, April, 2011.

[3]. Mohamad.H. Hassoun, “Fundamentals of Artificial Neural Networks”, A Bradford Book MIT Press, Cambridge, London, England,1995.

[4]. Avi Drissman, Dr. Sethi, CSC 496 ,February 26, 1997, “Handwriting Recognition Systems: An Overview”.

[5]. International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011 DOI :10.5121/ijcsit.2011.3103 27 “DIAGONAL BASED FEATURE EXTRACTION FOR HANDWRITTEN ALPHABETS RECOGNITION SYSTEM USING NEURAL NETWORK” J.Pradeep[1] , E.Srinivasan[2] and S.Himavathi[3] [1,2]Department of ECE, Pondicherry College Engineering, Pondicherry, India.

[6]. [3]Department of EEE, Pondicherry College Engineering, Pondicherry, India.

[7]. International journal of Emerging Trends in Science and Technology IJETST- Volume||01||Issue||03||Pages 405-410||May||ISSN 2348-9480 2014 “Handwriting Recognition System – A Survey” author Pooja Yadav & Ms. Neha Popli ITM University, Gurgaon .Corresponding Author Pooja Yadav ITM University.

[8]. Priya Sharma, Randhir Singh, Lecturer, Department of ECE, SAI Polytechnic, Badhani, Punjab, India Head, Department of ECE, SSCET, Badhani Punjab, India, International Journal of Engineering Trends and Technology- Volume4Issue3- 2013,” Survey and Classification of Character Recognition System”.

[9]. Jagruti Chandarana[1], Mayank Kapadia[2] , [1]Department of Electronics and Communication Engineering, UKA TARSADIA University , [2]Assistant Professor, Department of Electronics and Communication Engineering, UKA TARSADIA University, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 5, May 2014),” Optical Character Recognition”.
s

Asoke Nath "Handwritten English Character Recognition Using Edge Detection, Segmentation and Pattern Matching "International Journal of Engineering Science Invention (IJESI), vol. 07, no. 06, 2018, pp 60-66