

HANDWRITTEN CHARACTER RECOGNITION

Hirali S. Amrutiya¹, Payal P. Moghariya², Vatsal H. Shah³

¹Information Technology, Birla Vishwakarma Mahavidyalaya, amrutiyahirali@yahoo.in

²Information Technology, Birla Vishwakarma Maahavidyalaya, payal.moghariya@gmail.com

³Assistant Professor, Birla Vishwakarma Mahavidyalaya, vatsal.shah@bvmengineering.ac.in

Abstract— The paper Handwritten Character Recognition takes handwritten single character as input. The preprocessing is done first. To extract features of a character, Diagonal feature extraction is used. The chromosome bit generator is then used on the extracted features of the image to get a 378 bit length of chromosome.

Keywords- handwritten character recognition, preprocessing, Sobel edge detection, feature extraction, chromosome bit generator

I. INTRODUCTION

As the name suggest, handwritten character recognition are the techniques that are used to identify the handwritten characters. We are now in the era of technology. But even so, handwritten documents have made their way in this new era by making its presence felt online also. So now the handwritten characters can be classified as online and offline characters. On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. This kind of data is known as digital ink and can be regarded as a digital representation of handwriting. The obtained signal is converted into letter codes which are usable within computer and text-processing applications. Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting.

I. STEPS FOR CHARACTER RECOGNITION

The handwritten character recognition involves the following steps [1]:

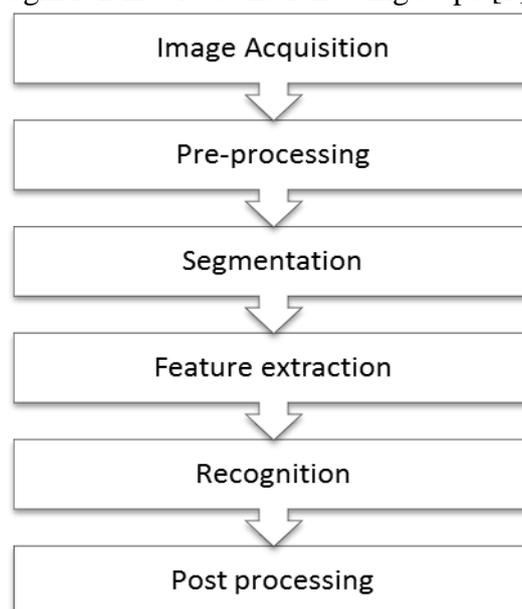


Figure 1. Steps for recognition

1.1 Image acquisition

Here, in this paper we are using algorithm for offline handwriting recognition. Thus first and foremost we require a scanned image as the input image. The format can be JPEG, BMP, JIF etc.

2.2 Preprocessing

This step involves a basic processing of the image before it is used for recognition by the system. It has to be processed in such a way that it is suitable for the system to understand. The steps involved are[2]:

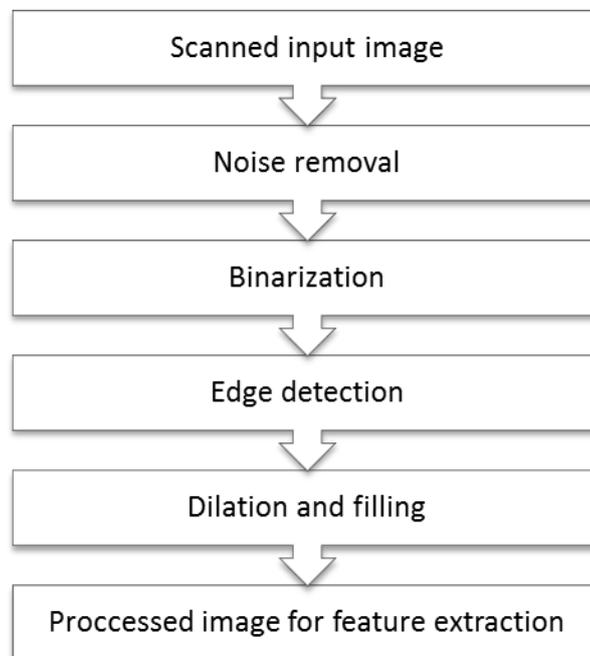


Figure 2. Steps for preprocessing

2.2.1 Noise removal

There are several reasons due to which noise gets added in the images. It could be from the mechanism that is used to acquire the image, the film grain or the electronic transmission of the image. The noise can be removed by linear filtering, median filtering or adaptive filtering.

2.2.2 Binarization

Binarization is a process where each pixel in the image is converted into one bit and assigned a value of 0 or 1 depending upon the mean value of all the pixels. If the value is greater than mean value then value assigned is 1 else 0. Thus it converts a greyscale document image to binary document image.

2.2.3 Edge Detection

Edge is a place where the intensity of the object changes drastically. Edges are often associated with the boundary of the objects in an image. Thus edge detection is used to identify the edges of an image. Edges are detected using Sobel method.

2.2.4 Dilation and filling

The basic effect of using dilation operator on binary image is to gradually enlarge the boundary of regions of the foreground pixels. Thus the areas of foreground pixels grow in size while holes within those regions become smaller.

2.2.5 Processed image for feature extraction

In this stage, the features of the characters that are crucial for classifying them at recognition stage are extracted. This is an important stage as its effective functioning improves the recognition rate and reduces the misclassification [3].

2.3 Segmentation

The first step is to segment or break the words, sentences to pieces such that a clear boundary is set between the characters. This is required as the algorithms for recognition can be applied to character only and not complete words. Also then each character is resized to $m \times n$ pixels.

2.4 Recognition

In the final step, the system tries to analyze and recognize the character that was given to it.

II. ALGORITHMS

2.1 Algorithm for pre processing

1. Start
2. `Img=imread('sample.jpeg');` for reading the image file
3. `Imshow(Img);` for displaying original image(refer Figure 3)
4. `Level=graythresh(mg);` for finding the mean level of the pixels
5. `Bnw=im2bw(Img,Level);` for converting the image to binary image
6. `Imshow(Bnw);` for displaying the binarized image(refer Figure 4)
7. `Isobel=edgw(Bnw,'sobel');` for finding the edges
8. `Imshow(Isobel);` for displaying the edge of image(refer Figure 5)
9. `S1=strel('line',3,90);` vertical structuring element
10. `S2=strel('line',3,0);` horizontal structuring element
11. `Idil=(Isobel,[s1 s2]);` for dilation
12. `Imshow(Idil);` for displaying dilated image(refer Figure 6)
13. `Ifil=imfill(Idil,'holes');` for filling holes
14. `Imshow(Ifil);` for displaying filled image(refer Figure 7)
15. End

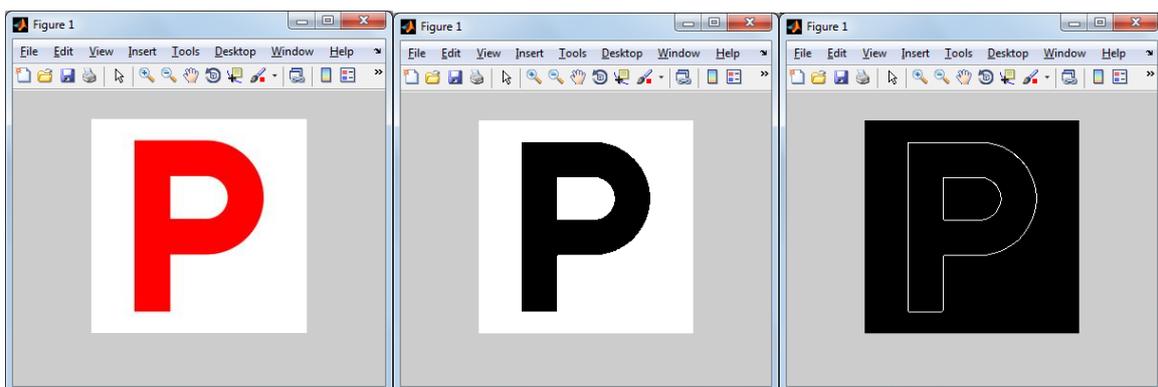


Figure 3

Figure 4

Figure 5

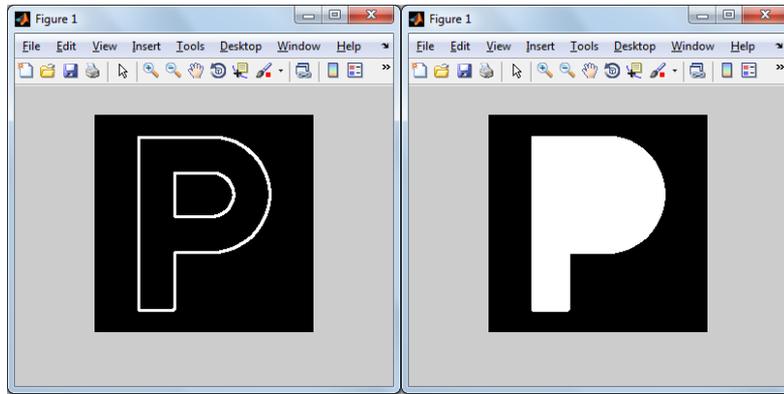


Figure 6

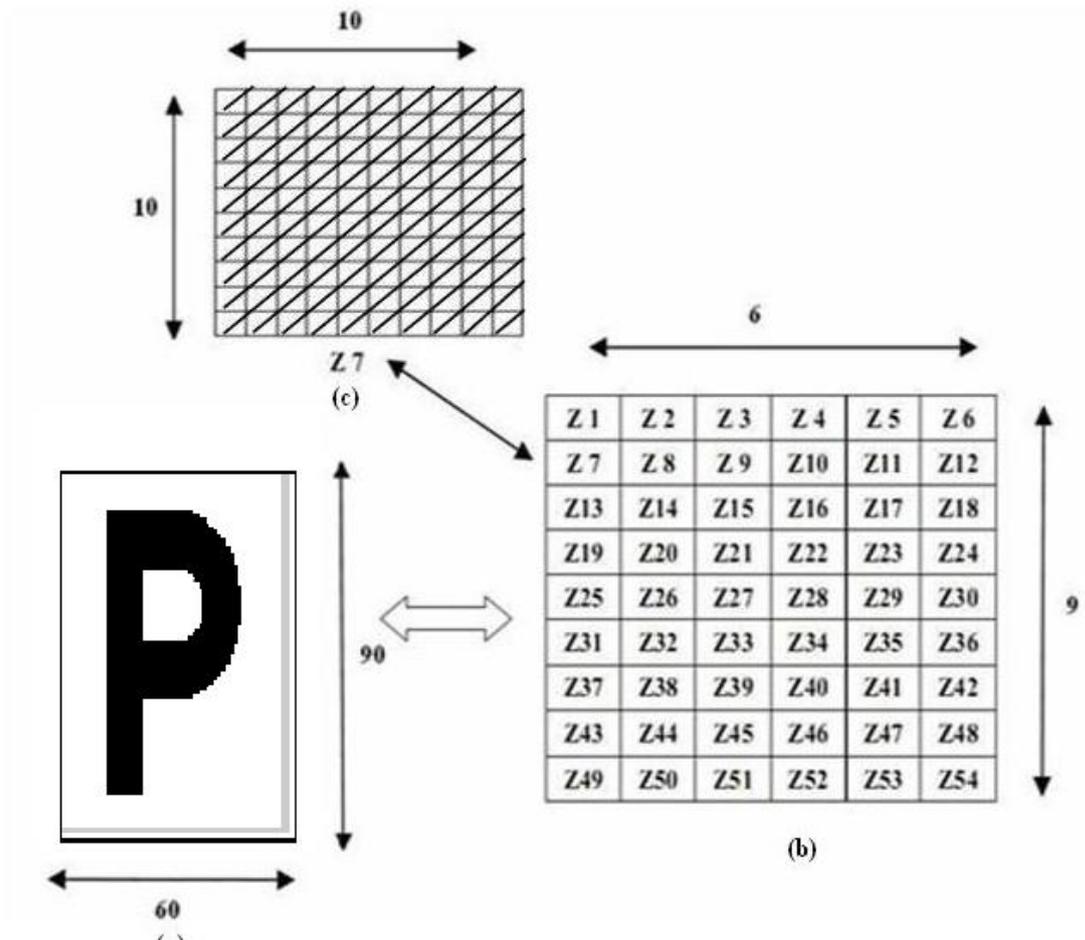
Figure 7

3.2 Algorithm for feature extraction

For feature extraction, first the image is resized to 90x60 pixels. Then the new image is divided into slots or zones each of 10x10 pixel size.

From each of the 10x10 sized blocks, the features are extracted by moving along its minor diagonal. The foreground (the pixels of color different than the background color) pixels are summed up along the diagonals. As in each of this 10x10 slot there will be 19 diagonals, we get 19 sub features which are averaged to form a single value for the entire zone. Thus, for a character there will be a total of 54 features each for a zone.

1. Start
2. Read image $I = \text{imread}(, \text{sample.bmp}')$;
3. Resize image into 90x60 $B = \text{resize}(I, [90 \ 60])$;
4. Find number of row and column $[c1 \ c2] = \text{size}(B)$;
5. Set $bs=10$; Block size (10x10)
6. Set $nob = (c1/bs) * (c2/bs)$; Total number of 10x10 block
7. Set $k=0$; $kk=0$;
8. For $i=1:(c1/bs)$
 - 8.1 For $j=1:(c2/bs)$
 - 8.2 $k=k+1$;
 - 8.3 $\text{Block}(:, :, kk+j) = B((bs*(i-1)+1:bs*(i-1)+bs), (bs*(j-1)+1:bs*(j-1)+bs))$;
 $\text{imwrite}(\text{Block}(:, :, kk+j), \text{strcat}(\text{int2str}(k), '.\text{bmp}'), '\text{bmp}')$;
 - 8.4 End
 - 8.5 $kk=kk+(c1/bs)$;
9. End
10. End (Dividing into Blocks and write each block into image) [1]



3.3 Algorithm for classification and recognition

In classification, the features of the image are analyzed. Basically, the classification is in two phases: training and testing.

In the training phase, the database is created for all the characters of the English language that is under consideration here. For each of the character, the features are mentioned. The features of all the characters should be different enough for it to be distinguishable.

In the testing phase, the unknown image's features are compared with all the images present in the database to find the correct and most nearer output.

First we create a chromosome function generator, which uses the binarized zoned image and generates a 162 bit length of chromosome. Each of the letters will have a unique chromosome.

The algorithm for this is:

1. Start
2. Clear global bitString;
3. Set global bitString;
4. For i=1:54
5. I=imread(strcat(int2str(i),".bmp"));
6. Set A=spdiags(I);
7. Set Avg=0;
8. Find Size of A [r c]
9. For j=1:c
10. For k=1:r
11. Avg=Avg+A(k,j);

