IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. II (May – Jun. 2015), PP 28-33
www.iosrjournals.org

A Review of Optical Character Recognition System for Recognition of Printed Text

Rajas Kiran Jambekar[1]
[1]*(Department of Computer Science, NMIMS University, India)*

Abstract: *As we move ahead in technology advancements, from simple data processing, to intelligent computing, one area of research undergoing advancement, is the system of reading text characters on an image. Optical Character Recognition (OCR) system is used for converting text characters on images into computer editable text characters. It includes steps such as Image acquisition, pre-processing of the image, segmentation of lines and characters, recognition of characters, and finally application. The image acquisition step determines the method for obtaining the image. Pre- processing of image includes enhancement of the image to make it suitable for recognition. Segmentation is the extraction of the character part of the image. Recognition is the comparison of the sample image with the template image. This paper focuses on the OCR system and includes information regarding the various operations that may be performed on the image for the recognition of characters.*

Keywords: *Image acquisition, OCR, Pre-Processing, Recognition, Segmentation, Text Characters.*

I. Introduction

Optical Character Recognition System is a relatively new field in the world of technology. Although research has been present on the topic since a few years, the topic has growing interest due to advancements in the processing capabilities of computers. OCR system has plenty applications in the field such as car number plate identification, processing cheques in banks, searching text in scanned documents, handwriting recognition, etc. [1].

Character Recognition involves conversion of text characters, on an image, into a form of text characters that can be edited by computers. The type of characters that are recognized by the system may include the basic alphanumeric characters including both uppercase and lowercase characters. The system may also be extended to recognize special characters and symbols.

The image used for the character recognition may contain printed text or handwritten text. This paper aims at providing a system which may be used for the recognition of printed text. The recognition of handwritten text is a difficult process due to the variations in the writing style in different individuals. Thus, the accuracy of recognition of handwritten text is relatively less than the recognition of printed text [1].

II. Methodology

The entire process of Character Recognition may be divided into five main stages:

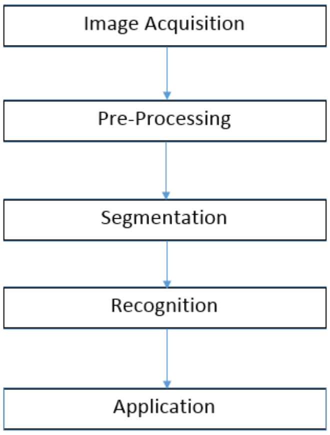


Fig. 1: OCR System Flowchart

Image Acquisition

Earlier, flat-bed scanners were used for obtaining clear images of standard quality and suitable for recognition. Using a scanner has benefits such as low noise levels, virtually no blurring and low text skewing.

However, the working of the flat-bed scanners is slow and it requires physical connection with computers which limits the process.

However, due to the expansive growth of technology, we now have high grade digital cameras on smaller devices such as mobile phones. The image quality obtained using mobile phones is good enough for processing of the image. The only issues while capturing images using a digital camera are shadows, blurring and text-skewing. However, up to a certain limit, these issues may be managed using image enhancement techniques discussed in the next section.

Pre-Processing

Pre-processing of the image includes modifications to the image to make it suitable for recognition. A typical OCR system may use the following techniques for image enhancement:

a. RGB to Gray-Scale Conversion

The image acquired for recognition is usually a colour image, i.e. the pixel value consists of the combination of the three colour components, red, green and blue. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix.

Detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image.

b. Skew Correction

Camera captured images may suffer from skew and perspective distortion [2]. As discussed above, this effect is due to improper image capturing technique. The horizontal text axis may suffer rotation at some degrees. Such an effect may be reduced by rotation of the image at a certain degree. The calculation method for rotation of the image has been described in the article [3].

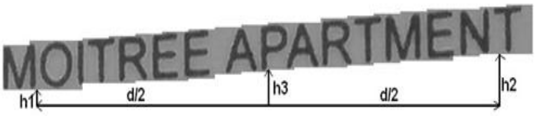


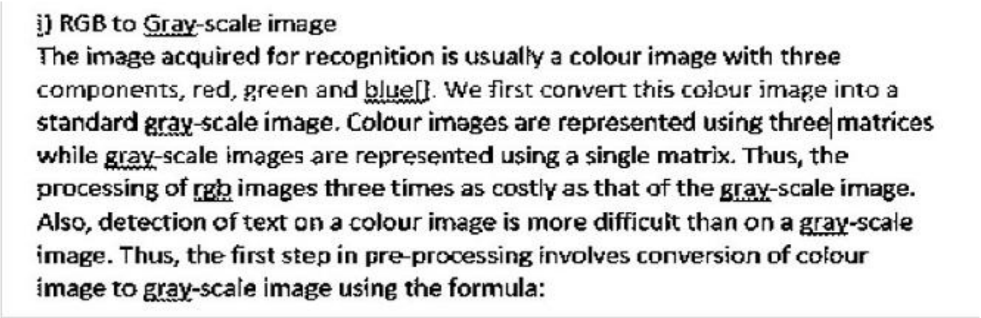
Fig 2: Skewed image [3]

c. Binarization

Binarization is the process where a threshold is selected for conversion of pixel values into 0s and 1s. The black pixels are represented by a 0 and the white pixels are represented by 1. The threshold value may be selected using various methods. A simple way to determine the threshold value would be to find the median value of the maximum and minimum intensity values in the image.

It would be represented as:

Threshold value = (Imax + Imin)/2



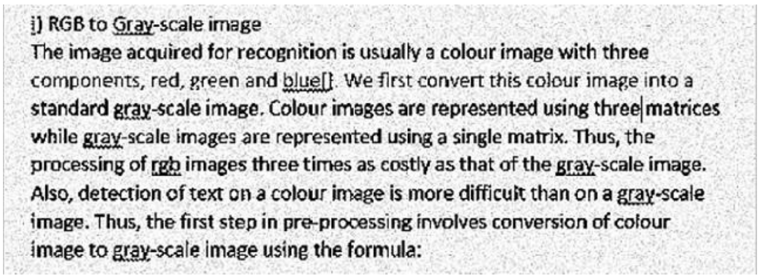
i) RGB to Gray-scale image

The image acquired for recognition is usually a colour image with three components, red, green and blue[]. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix. Thus, the processing of rgb images three times as costly as that of the gray-scale image. Also, detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image using the formula:

Fig 3: Binarized image

d. Noise Reduction

Noise is unwanted pixels present in an image. They may be in the form of Salt and Pepper noise or Gaussian noise. We use the low pass filter for filtering the Gaussian noise from the image. Since the presence of salt and pepper noise is not as high as the presence of the Gaussian noise, we do not filter it.



i) RGB to Gray-scale image
The image acquired for recognition is usually a colour image with three components, red, green and blue[]. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix. Thus, the processing of rgb images three times as costly as that of the gray-scale image. Also, detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image using the formula:

Fig 4: Noisy Image

e. Thinning

Thinning is the process of reducing the width of the foreground pixels. While thinning, it is necessary to maintain the form of the characters on the image. Thinning is done on the basis of neighborhood of a pixel. e.g. If a line on an image is of 3 pixel width, the thinning function will change the border pixels of the line and the output image will consist of line of one pixel width [4].

Segmentation

Once the image is enhanced, it is passed to the segmentation module where each character is separated from the other. The Image at this point may be divided into two types of regions, background region and foreground region [4].

This module is thus responsible for the separation of the foreground region from the background region. The foreground region is a collection of text characters placed on the same line as well as on different lines. The segmentation process works in two steps:

1. Line Segmentation
2. Character Segmentation

Line segmentation is the separation of the different lines of characters present in the image. Each line is defined by a minimum vertical gap between the characters present on a line and on the line above and below it. This gap can be used for the detection and separation of different lines of characters.

Character Segmentation is the separation of characters present in the same line. Once the lines are separated, each character is extracted from the line. There is a constant horizontal gap between characters which is used for the separation of characters. Thus, images corresponding to individual characters are extracted which are fed to the recognition module.

Recognition

This is the final step of the process of recognition of characters. There are various methods available for comparison of images, such as image correlation, feature extraction and comparison, chain code comparison, artificial neural networks, etc. Image correlation method is the direct correlation function used on the sample image and the template image. Feature extraction is a process of analyzing the sample image and deriving specific features from the image. The comparison of these features with the features of template images is used for recognizing the character. Chain codes are code sequences generated for each image based on the neighborhood of pixels. The longest common sub-sequence is found between the code of sample image and the template images which decides the output character.

Each of these techniques requires the use of stored template images used for comparison. The output after the comparison is an editable text character which is added to a buffer. When all the characters of a word are filled in the buffer, the string is passed to the application for further processing. However, this can also be done after the recognition of each line of text or after the recognition of all the lines of text in the image.

Application

The output obtained from the recognition module is in the form of editable text characters i.e. ASCII characters. These characters are initially stored in a buffer from which they may be stored in a file or further processed to derive certain output such as text to speech conversion, business card to contact conversion, etc.

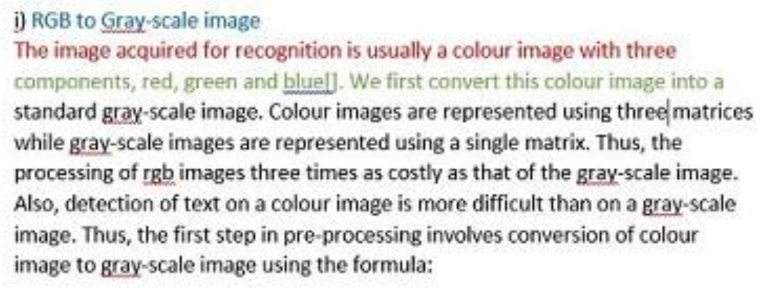
III. Proposed System

The described system has modules which can be implemented using a variety of algorithms. This paper presents the following algorithms for implementing some of the modules.

RGB to Gray-Scale Conversion

Conversion of image in to gray-scale can be done using the following formula. The new pixel value is computed for each pixel.

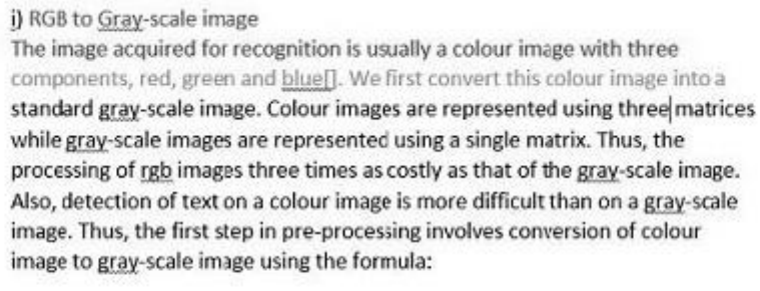
f(x,y) = 0.299 x r(x,y) + 0.587 x g(x,y) + 0.114 x b(x,y)



i) RGB to Gray-scale image

The image acquired for recognition is usually a colour image with three components, red, green and blue[]. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix. Thus, the processing of rgb images three times as costly as that of the gray-scale image. Also, detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image using the formula:

Fig 5: Colour Image



i) RGB to Gray-scale image

The image acquired for recognition is usually a colour image with three components, red, green and blue[]. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix. Thus, the processing of rgb images three times as costly as that of the gray-scale image. Also, detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image using the formula:

Fig. 6: Gray-Scale Image

Skew-Correction

This paper describes a slightly modified method for skew correction. The text regions in the referred article [3] de-skews each text region separately by repeated calculation of skew angle. The process requires repeated calculation of skew angle and image rotation for each extracted text region. This makes the algorithm more computer intensive. A slight modification to the skew correction system is thus proposed.

The text present in the image has more tendency to suffer from the same degree of skew throughout the image. The system scans the text region in the image from the top and the bottom and determines the skew angle twice. Then, taking average of both the skew angles, the image is rotated by the calculated degree.

Since the process of rotation of the image is performed only once for the entire image, it proves to be more efficient.

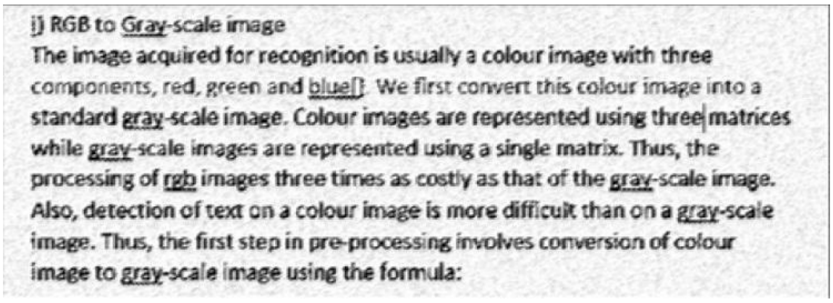
Noise Reduction

Given below in the figure is the low pass filter mask. It is basically used for removal of Gaussian noise from the image. The image suffers from slight blurring effect, thus, the selection of the size of the mask is crucial.

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Fig 7: Noise Reduction Mask

Applying the mask on a noisy image provides a noise reduced result with a slight blurring effect.



i) RGB to Gray-scale image
The image acquired for recognition is usually a colour image with three components, red, green and blue[]. We first convert this colour image into a standard gray-scale image. Colour images are represented using three matrices while gray-scale images are represented using a single matrix. Thus, the processing of rgb images three times as costly as that of the gray-scale image. Also, detection of text on a colour image is more difficult than on a gray-scale image. Thus, the first step in pre-processing involves conversion of colour image to gray-scale image using the formula:

Fig 8: Noise Reduced Image

Binarization

With varying images, using averaging method to determine threshold value may not produce the best results. Having referred to different methods for thresholding, we have selected the Ostu's Method for determining the threshold value. Ostu's method for thresholding is based on formation of two distinct classes of pixels, one which contains background pixel intensities and the other containing foreground pixel intensities [5].

Segmentation

The segmentation process may be performed using methods such as blob colouring, peak-to-valley method, etc. [6]. To separate individual characters present in the foreground region, the process of segmentation is divided into two steps:

a. Marking

Marking is the process where the boundaries around a character are marked and stored, before it is segmented. For each line of text in the image, the beginning and the ending rows and columns of the pixels corresponding to each character in the text line are marked.

We have analyzed various methods which involve character segmentation without dedicated marking of boundaries around the image. The method of marking the boundaries has a positive effect of improving the performance of the system by performing similar task continuously for each text line. The process of marking is described below.

Line marking is responsible for marking the upper and lower boundaries of each line of text present in the image. This can be done by by first marking the top boundary of the text on the image by checking pixel values along the rows starting from the top of the image. If a foreground pixel is found, the pixel position is marked and added to the topForeground list. Similarly, for marking the bottom boundary of the line, the pixel values are checked along each row. If no foreground pixel is found along a row, the row above the current row is marked and it's position is added to the bottomForeground list.

Using this method, two lists are obtained. One indicating the top boundary of the lines and the other indicating the bottom boundary of the lines. If an image contains 'n' text lines, the length of both lists is 'n'.

The process of character marking is similar to that of line marking. The upper and lower boundaries for each line of text in the image have been marked. Now, the start and the end boundary column for each character along a line are marked. The process involves vertical scanning of each text line from left to right and checking for foreground and background pixels.

Similar to line marking, we scan the image from top to bottom along all columns starting from the top of the line up to the bottom of the line. If a foreground pixel is found in a column, the pixel position is marked and added to the leftForeground list. Continuing the scanning along the columns, if a column is found without any foreground pixel, it indicates the end of the character and thus we mark the pixel position and add it to the rightForeground list.

Thus, we obtain the boundaries for each line and character along the lines.

b. Extraction

The system now extracts each character at a time by using the four boundaries and store it as a separate image 'sample'. The sample has boundaries that exactly match the first and last pixel along each direction. Thus, there is need to perform the additional step of boundary selection [6].

However, before the sample can be passed to the recognition module, the size of the sample must be adjusted with respect to the size of the template images. This is necessary since the recognition module can only compare images of the same size. The image is thus resized to the size of the template images. We maintain templates of a small size to ensure that the resizing of sample always decrease the dimensions thus, preventing blurring of image. However, the size must also not be so small that fine details of the sample are lost. Thus, a size of no less than 20px is selected for template.

The above algorithm is used for only a single line of text at a time. Once the characters in a line have been marked, the character marking system pauses. The list of leftForeground and rightForeground values along with the top and bottom boundary of current line is passed to the extraction module.

The system only restarts when all the characters in the current line have been processed by the following modules. When the system restarts, the next value of top and bottom boundary are selected and the corresponding characters on the line are marked. This process is repeated for each line in the text region.

Recognition

We have selected image correlation for recognition of sample image with the template image. The sample image is compared to each of the template images using the correlation function. The function returns a scalar value with a value ranging between 0 and 1. Higher the value indicates better correlation between the sample and the template images. Thus, we determine the character on the basis of the best value of correlation. The recognized character is added to an output buffer string.

IV. Conclusion And Future Scope

We have reviewed the process of recognition of characters present on an image. This paper presents a system which may be implemented as is or can be further enhanced. The current system is good enough for recognition of simple characters and numbers along with the detection of white spaces. However, it may be further extended to support text characters with a variety of fonts and special symbols. The use of neural networks and adaptive learning for recognition may enhance the recognition capabilities of the system extensively. The possible applications for this OCR system may be in the form of license plate recognition, business card to phone contact conversion, document image to editable document conversion.

References

[1]. Disha Bhattacharjee, Deepti Tripathi, Rubi Debnath, Vivek Hanumante, Sahadev Roy. “A Novel Approach for Character Recognition”, International Journal of Engineering Trends and Technology (IJETT) – Volume 10 Number 6 - Apr 2014

[2]. Ayatullah Faruk Mollah, Nabamita Majumder, Subhadip Basu and Mita Nasipuri. “Design of an Optical Character Recognition System for Camera Based Handheld Devices”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011

[3]. A. F. Mollah, S. Basu, N. Das, R. Sarkar, M. Nasipuri, M. Kundu, “Text/Graphics Separation and Skew Correction of Text Regions of Business Card Images for Mobile Devices”, Journal of Computing, Vol. 2, Issue 2, February 2010, ISSN 2151-9617

[4]. Ravi Kumar, Anurag Anand, Nikunj Sharma, “Recognition of English Characters by Codes Generated Using Neighbour Identification”. International Journal of Application or Innovation in Engineering and Management (IJAIEM) – Vol. 2, Issue 4, April 2013, ISSN 2319-4847

[5]. N. Otsu, "A threshold selection method from gray level histogram”, IEEE Transactions in Systems, Man, and Cybernetics, Vol. 9, pp. 62 – 66

[6]. Ghugardare, Rakhi P., Sandip P. Narote, P. Mukherji, and Prathamesh M. Kulkarni. "Optical character recognition system for seven segment display images of measuring instruments." In TENCON 2009-2009 IEEE Region 10 Conference, pp. 1-6. IEEE, 2009.