International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Implementation of Handwritten Character Recognition using Template Matching

Ayushi Chudgor†* and Vinaya Sawant†

†Department of Information Technology, DJSCE Mumbai University, Vile Parle, Mumbai -400 057, India

Accepted 14 Nov 2016, Available online 18 Nov 2016, **Vol.6, No.6 (Dec 2016)**

Abstract

In the recent years, handwritten character recognition has been immensely researched and several methods have been implemented. In this paper, a system for recognition of handwritten code by Template Matching is presented. For this purpose, I have build my own database of handwritten characters in various fonts and size, which includes lower case letters, numbers and selected special characters. The pipeline is built in MATLAB and works taking by an input image, pre processes the input image and recognises the characters from the trained data to give an output of the given code as printed text.

Keywords: *Optical Character Recognition, Template Matching*

1. Introduction

In the recent years, handwriting recognition has become an active and important area of study due to increase in clamor for the recognition of text from historical documents and digitizing them for their preservation. It has application to many fields such as vehicle license plate recognition, bank cheque processing, digitization of historical documents, signature analysis etc.; It is difficult to implement as a single character can be represented in various ways, which increases the complexity during recognition of character. Optical Character recognition can be abbreviated as OCR refers to the process of conversion of images of handwritten or printed text to machine-readable text. There are majorly two types of optical character recognition namely: (i) Offline Character recognition (ii) Online Character recognition. Online Character recognition refers to characters being processed when it is under creation. There are various online character recognition tools available on internet such as apprise java OCR, free-ocr.com etc., While in offline character recognition, a document needs to be generated, digitized, stored in computer and finally processed. This images used for offline character recognition needs pre-processing to get it ready for further processing by OCR. The techniques used in pre-processing are dataset acquisition, binarisation, removal of noise, and line and character segmentation. The output of character wise segmentation is then compared against database of templates to generate the required output text. The observations and results achieved have been discussed in this paper.

*Corresponding author: **Ayushi Chudgor**

2. Block Diagram of proposed Implementation

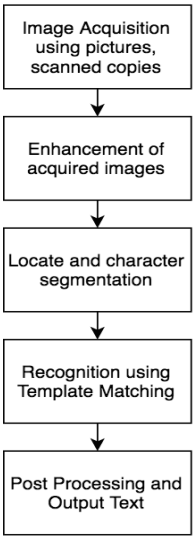

Fig.1 Steps involved in generation of printed text from an image

The proposed system works as follows: an image or scanned copy of handwritten code is first acquired. The image is processed using various image processing techniques like binarisation, removal of noise etc., so that it can it can be used for further processing. The image received is segmented horizontally and vertically to obtain individual characters of the image. The characters are matched with the templates provided to recognize the specific character. After recognition and matching, the character is generated in the printed text file provided as output. In case of

mismatching in templates and characters, manual processing is carried out to generate the required text.

3. Pre-processing Phase

3.1 Dataset Acquisition

Dataset contains 30 samples of handwritten text received from 4 people that have been scanned by a scanner to obtain best quality scan for template dataset. The data set includes lower case letters and selected special characters.

3.2 Binarization

The scanned images of texts need to be pre-processed so that they are in suitable form for character recognition. This means we first need to convert RGB or gray images to black and white images, this process is known as binarization. I used the 'rgb2gray' function in MATLAB to first convert the RGB image to gray scale image. Followed by the used of 'im2bw' function that converts the gray scale image to black and white image by choosing an appropriate threshold value.

The process of binarization is illustrated through the following images where:

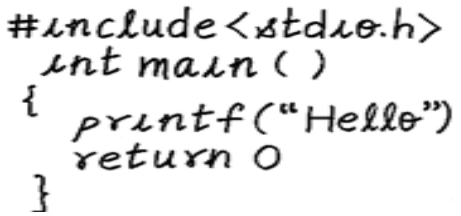

Fig.2a Shows the result of conversion original input image to gray scale image.

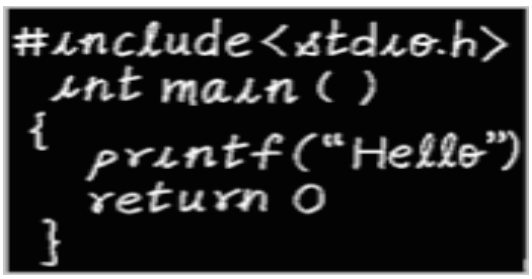

Fig.2b Shows the result of conversion of grey scale image to black and white image.

3.3 Removal of noise

To remove unwanted noise from the black and white images, I used 'bwareaopen' function in MATLAB to remove all small components below 30 pixels from the image. The removal of smaller components is an important step as it will help remove the unwanted noise in the image which if not removed can ultimately affect character wise segmentation.

4. Segmentation Phase

4.1 Line Segmentation

To achieve the output in similar format as the input image, it is necessary to segment the image first horizontally as it helps maintain the order of the characters. Here, a basic assumption is made of no overlapping between the lines and sum of the pixels in a row is zero. This factor is also considered to be the break between the lines of text in the image. Using the 'clip' function, we segment the text row wise and store the remaining lines of text in other matrix so that the text is not lost or overlapped before it can be segmented.

Fig.3 Horizontal Line segmentation

4.2 Character Segmentation

For the template matching method to work, it is necessary for us to segment the output of line wise segmentation further to single character wise segmentation. I used the approach of 'bwlabel' to check the connectivity of the letters and label the connected components. The next step is to crop each labeled group of pixels by finding the group's minimum and maximum values of its row and column and extracting the letter out. I resized the cropped image to 42 x 24 pixels; to better facilitate template matching as the templates stored in the database are also in the size of 42 x 24 pixels. Resizing helped in achieving sharper boundaries for the segmented characters.

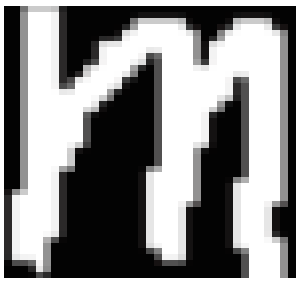

Fig.4a Before Resizing **Fig.4b** After Resizing

5. Optical Character Recognition

To recognize the characters from the input image is carried out using the Template Matching method. After the character wise segmented images are received, matching them with the templates of the dataset identifies the characters. This is done with the help of 'corr2' function of MATLAB. The concept of this function is to detect similarities in 2D patterns with cross correlation method. Here, the input image is stored as matrix in A_{mn} while compared against a

template, which is in B_{mn}. The return value 'r' indicates the match ability between the input image and template. After all comparisons, the highest correlation coefficient value of 'r' is identified as the lower case letter or selected special character.

'corr2' function in MATLAB computes the correlation coefficient using the equation as follows:

$$r = \frac{\sum_m \sum_n (A_{mn} - A)(B_{mn} - B)}{\sqrt{\left(\sum_m \sum_n (A_{mn} - A)^2\right)\left(\sum_m \sum_n (B_{mn} - B)^2\right)}}$$

6. Results

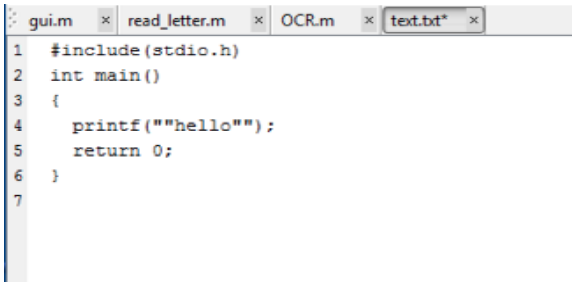

Fig.5 Result obtained in text output

The result obtained in generated text file is correct owing to the line and character wise segmentation method, which starts to read the input from the first line of the texts to the bottom line and then from left side to the right side. The advantages of using this method is owing to its simplicity but can be used for recognizing various sets of handwriting provided through templates.

Table 1 Experiment

S. No	Phase	Accuracy
1	Pre-processing	100%
2	Segmentation	91%
3	Handwritten code Recognition	73%

The major factor that affected the accuracy of handwritten code was the problem in mismatch in templates and the character of input image. The dataset of the templates was comparably small and was not able to handle variations in similar looking letters such as 'i' and 'j'. Also, in pre-processing phase, there were instances that removed the dot on top of 'i' and 'j'. These caused problem in template matching as it was often mismatched with 'L' of the dataset.

But, considering the all factors involved it is an easy and efficient method to implement and learn the concept of machine learning.

Conclusions and Future Scope

1) Optical Character recognition project using template matching has been implemented successfully.
2) The importance of pre-processing phase in this project is immense and includes steps like binarization, removal of smaller components and horizontal and vertical segmentation.
3) The process of template matching thus can be summarized as comparing the input image with a stored template and identify the character from the input image according to its highest match using coefficient correlation. Through this method, handwritten text is recognized with good accuracy.
4) The scope for future work is immense. As here the images were scanned for better resolution. Inclusion of advanced image processing techniques like skew correction, adjustments for lighting in image could be done.
5) The accuracy of the project can be improved when it can be expanded to include a wide variety of handwritings and bigger set of handwritings which enables us to encompass large set of variations.
6) With inclusion of larger data sets and handwriting variation, advanced techniques like neural networks could also be implemented.

8. References

Rachit Virendra Adhvaryu, (2013), Optical Character Recognition using Template Matching (Alphabets and Numbers), *International Journal of Computer Science Engineering*, Vol. 3, 227-232.

Lipi Shah, Ripal Patel, Shreyal Patel, Jay Maniar, (2014), Skew Detection and Correction for Gujarati Printed and Handwritten Character using Linear Regression, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, 642-648.

Jatin M Patil, Ashok P. Mane, (2013), Multi Font And Size Optical Character Recognition Using Template Matching, *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, 504-506.

Sandeep Tiwari, Shivangi Mishra, Priyank Bhatia, Praveen Km. Yadav, (2013), Optical Character Recognition using MATLAB, *International Journal of Advanced Research in Electronics and Communication Engineering*, Vol. 2, 579-582.