OFFLINE HANDWRITING RECOGNITION USING NEURAL NETWORKS

Harsha K C, Satya Prakash
harsha.kc@iiitb.org,satya.prakash@iiitb.org

Abstract

This is a report summarizing the approach taken to solve the problem of handwritten character recognition. The problem statement is similar to any OCR system, whereas the input here is an image of normal handwritten text.

Keywords: handwriting-recognition,character segmentation, zoning, neural network

1. INTRODUCTION

This problem belong to the OCR category which has been solved for digits and printed characters. There are very few solutions which can do a generic handwriting recognition. The report one such method chosen to tackle this problem. Handwriting recognition can again be seen as two types: Offline and Online Handwriting recognition. Online problem category is considered less harder as the stroke lines and start and end points could be tracked easily. The offline category is more of an image processing problem to derive meaningful representations for each character and build a model which could be used for recognition.

2. Problem Background

Handwriting recognition is not so trivial as individual character recognition due to the following reasons. The individual OCR input would be an already available single character where as, here it would be contiguous writing and the major task is to split the words into individual characters.

Next problem is each person's handwriting is different and text taken over a paragraph written by the same person will in turn have variations for the same character. The generic handwritten character recognition, that is to solve the problem making it writer independent, as present day systems for digits is a bigger problem to solve. The first step in solving this problem is to solve the problem assuming we have to identify characters from a given person , by providing a fair amount of training data. Once this problem is solved, the training corpus accumulated from many people can then be used to build a good representative model for each character.

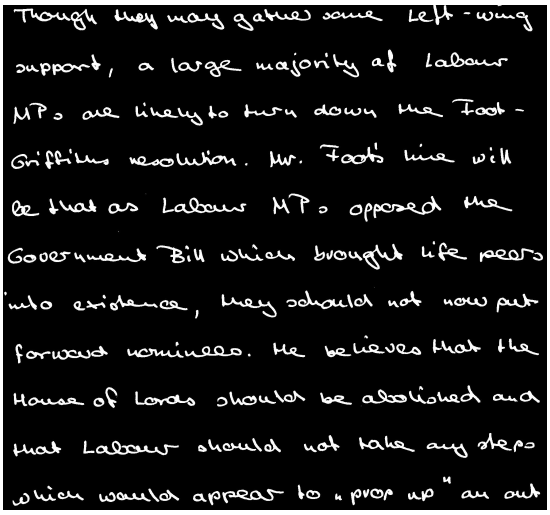

Figure 1. Sample Image used for Handwriting recognition

3. Description of the Approach

The IAM Handwriting image database was used for this purpose. In general any grayscale image could be fed as input to this system considering the lines are well spaced. Each step taken in this process explained below.

- First task was to identify each line in the text and separate them. This is a simple task as one has to identify the troughs in the horizontal histogram

plot. These troughs would ideally correspond to the spaces between the lines and could be split to different images to obtain lines.

Figure 2. First Line extracted

- Next task was to identify words in each of these lines. This task again is similar to using a split based on histogram as above, but the splits would be in the vertical direction.

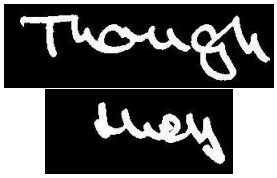

Figure 3. Words extracted from above line

- After this comes the tricky part of obtaining individual characters from each word. An edge detection algorithm was used one each word to obtain a gradient matrix. There are many standard filters available, the one used here was a Sobel filter. It was interesting to see that the histogram of this gradient matrix had candidate peaks and troughs which could be used to approximate the character width for split. Furthermore, the gradient of the gradient gave a smoother crests and troughs which could be used for character segmentation.

  There are many candidate peaks and troughs in the histogram and thus we need a mechanism to decide which of these could be potential segmentation points. In the training phase, we made use of the word length as a parameter to decide candidate peaks. The idea was simple, if a word had n characters, we would need n-1 splits to obtain each character. Thus based on the image size, the initial split points were decided based on the ratio of image width and number of splits needed. Based on this location, a trough is identified which is near to this ideal split location.

  This approach would work reasonably well based on the assumption that for a person, the character width doesn't change by a large factor across words. So throughout the training phase, each split size is kept track and the mean character width

Figure 4. Character Segmentation

  could be used as an approximation to split the words in the testing phase.

- After the character segmentation, the next stage is to build a model for each letter. Again many approaches could be used here. The most basic Zoning approach was chosen here. The image after segmentation into individual character was padded with background cells to make the image a square image.

  The entire image is divided into a 4×4 matrix, and each zone's average intensity is computed. Together the 16 zones contribute as a feature vector.

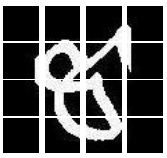

Figure 5. Zoning

4. Neural Networks

As any other OCR task, neural networks were used to build a model for the handwriting recognition as well. The feature vector obtained from the above process has a dimension of 1×16. For each character in the word a similar feature vector is obtained. The model should be representative of the individual's handwriting to get better results, that is, most possible variations of letters and all letters should be obtained for the training phase. A simple exercise could be to ask the individual to write a sentence such as *" The quick brown fox jumped over the lazy dogs "* which has all 26 letters. Writing this line repeatedly should capture the person's variations in characters.

For each feature vector, a corresponding label is created which has dimension of 1×26 and has a 1 indicating which letter is it. This data is then fed to a neural network and trained. The data could be split for train and test appropriately.

5. RESULTS

Owing to the errors in the segmentation process which could be lossy at times, 90% accuracy was obtained after the testing phase. This could be tremendously improved if the segmentation process is tweaked or a better feature vector could be used.

6. FUTURE WORK

One novel approach to the feature vector is to make use of piecewise cubic Bezier curves to represent the characters. A simple edge detection algorithm followed by corner detection would give points of interest on the character. If a curve could be fit to this model using Bezier curves, it could serve as a more robust feature for each character.

7. ACKNOWLEDGEMENTS

We would like to take this opportunity to thank Enrico Schroder whose prior work of Writer Identification, helped us getting started with line and word segmentation approaches.

Also, we would like to thank Prof. G.Srinivasan Raghavan for his support and guidance throughout the project.