Handwritten Devanagari Character Recognition

Introduction

This project has been taken as a part of fulfillment of the syllabus of B.Tech, Computer Science and Engineering. This section majorly deals with the background and justification of the problem, areas where the solution can be applied and the approach taken to handle the problem.

Abstract

Optical Character Recognition is a procedure to recognize handwritten or printed characters from documents, automatically by machine, to transform them into digitally editable documents. The problem of Optical Character Recognition (OCR) can be considered to be subdivided into two major parts:

- Segmentation of the document
- Character recognition.

The main focus of this project is to work on the area of character recognition using a cognitive approach, which is both simple and less resourceful than other techniques already in use for the same purpose centering on neural networks and expert systems.

In the project, the document has been assumed to be scanned or photographed so as to generate a JPEG image file of the character to be recognized. Algorithms have been developed to address the problems of recognition of an appropriate threshold to differentiate between foreground pixels and background pixels, binarization of the image, noise removal, thinning, scaling, trimming the whitespaces, statistical and momentum-based feature extraction and storage of those extracted features.

An algorithm has been developed to identify an appropriate threshold value for each document dynamically depending on the exact values of its pixels. This threshold value helps to differentiate between the foreground and background pixels of the document. The next stage employs a simple and obvious logic for binarization of image by identifying the foreground and background pixels by comparing their values with that of the threshold found in the previous step and representing them by ‘1’ and ’0’ respectively.

In this project, for noise removal simple algorithms are used. We have restricted ourselves to one-pixel and two-pixel based noise. In the thinning phase, an algorithm is used to remove redundant pixels from the image. For this purpose, we have also used a mask which produces zero redundancy in the image. We have used algorithm to scale up/down the image. However here it is used to produce an image of the size 30x30 pixels. The size helps in logical division of the image which is needed for the feature extraction.

In the feature extraction phase, we worked on two types of features:

1. Statistical Feature: no. of horizontal zero crossing, no. of vertex points, no. of joining points, position of vertical bar etc.

2. Momentum-based Feature: average no. of black pixels, skewness, kurtosis & normalized skewness & kurtosis of the image.

Extraction of these features and storing them in a database helps to uniquely identify the character based on some recognition procedure as used in neural network.

The result obtained is satisfactory considering time and resource constraints, but needs refinement in some of the areas as mentioned in details in the following text. Verification on a larger set of samples and the corresponding statistical analysis, if can be taken up as follow up, is likely to generate more refinement and bring out more issues that can be solved then.

Definition, acronyms, standards and abbreviations

- **Background pixel:** the pixel that represents a part of the background (non-written part) of the hand-written document
- **Binarization:** conversion of the image of a handwritten document into a format of '0' and '1' where 0 represents background pixels and 1 represents foreground pixels.
- **Character**: a single alphabet in the hand-written document
- **Foreground pixel**: the pixel that represents a part of the written areas of the hand-written document.
- **Image**: the impression of the hand-written document or a part of it after segmentation taken in the JPEG format.
- **Noise**: corrupted values of pixels which does not represent the actual information about the image.
- **Pixel**: the smallest indivisible part of an image.
- **Threshold**: a grayscale value that marks the line of distinction between the range of grayscale values for foreground and background pixels.
- **Header-line**: an important feature of the Devanagari script usually called "Matra".
- **Zone**: image is divided into 3x3 logical matrixes. Each room of the matrix is known as a zone.
- **Vertex Point**: It is the open-end point(s) present in the image.
- **Junction Point**: It is the point where two or more than two line intersects each other.
- **Vertical bar**: It denotes the lines parallel to conventional y axis.

Technologies used

- The project has been developed on Java™ platform using J2SDK 1.6 standard by Sun Microsystem Inc.
- The IDE used for the purpose is Eclipse.
- The GUI has also been developed using Swing framework in Java 1.6.