ISSN:2231-0711
www.ijcset.net

Text Extraction from Visiting Card

Amol Darekar[#1], Vaibhav Ghatol[#2], Saurabh Karle [#3] , Anita Thengade [#4]

[#]*Department of Computer Engineering, MITCOE, Pune University*
Pune, India
[1]amoldarekar.p@gmail.com
[2] vghatol@gmail.com
[3] saurabhkarle05@gmail.com
[4] anita.thengade@mitcoe.edu.in

***Abstract*— This paper describe the one of the method of text extraction from visiting card images with fanciful design containing complicated color background and reverse contrast regions. The proposed method is based on edge detection followed by grouping of edges based on size, orientation and color attributes. After detecting edges of text from card we can apply image processing steps such as noise removal, segmentation to extract text from complex image. This text image can be recognized using OCR .We can design OCR using Artificial Neural Network. Some post processing like color identification and binarization will be helpful to get a pure binary text image for OCR.**

***Keywords*— OCR (Optical Character Recognition), Binarization, Segmentation**

I. INTRODUCTION

Recently there are various application of document engineering is found in visiting card which readily capture visiting card images followed by optical character recognition (OCR) to build a visiting card database. The application provides for document information portability, thus avoiding the need to carry a large number of visiting cards and facilitating retrieval of visiting card information from the database. While gaining its popularity, the application faces an obstacle to its full potential due to fanciful designs that are becoming common among visiting cards. Two main problems encountered are the complex background that contains texture or highly varying color and the presence of reverse contrast text regions. In this paper, we tried to recover this with two issues.

To address the above issues, we first surveyed the literature to find any existing methods for text extraction. The more straightforward approaches are the thresholding algorithms [1, 2, 3]. In [1], several single-stage thresholding algorithms are studied using either global or local thresholding techniques. Multi-stage thresholding methods are proposed in [2, 3] where a second stage thresholding based on the result of the first stage is done to enhance the result. Thresholding techniques are efficient but generally they assume that the text has a darker color than the background. For visiting cards that contain regions of reverse contrast, these algorithms failed. Graphical foregrounds are not considered in these algorithms either. Pietikainen and Okun [4] use edge detectors to extract text from grey scale or color page images. In their method, a gradient magnitude image obtained from the original image is divided into a grid of blocks. The blocks are classified as text block or non-text block based on the total number of edges in the block. The method fails in extracting larger size text and erroneously treats graphical foreground as text because of the large amount of edges in the texture blocks. For visiting cards which have a variety of text sizes and graphical foregrounds, this method performs poorly. The problem of reverse contrast text areas remains unsolved. In [5], Strouthopoulos et al propose a solution for locating text in complex color images. An unsupervised Adaptive Color Reduction is used to get principal colors in the image. For each principal color, a binary image is created and an RLSA is used to form object blocks which are then classified as text blocks or non-text block based on the block characteristics. All the text areas are merged in the final output. Though the method is able to handle complex color text with complex color backgrounds, it recognizes only horizontal long text lines with little space in between characters. Moreover, this method is slow when there are many colors in the image.

II. PROPOSED METHOD

Our proposed method is described with the help of following flowchart.

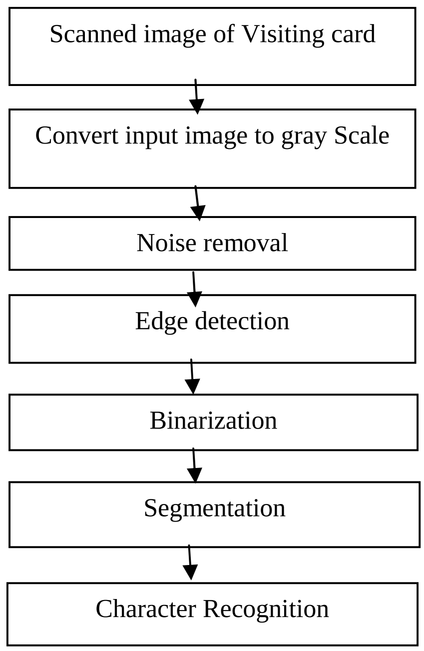


Fig1. System overview.

A. Gray Scale Conversion

Size of an input image varies depending upon the resolution scanner or size of card. Next, it is converted to 8-bit grayscale image using the formula $G = 0.299*R + 0.587*G + 0.114*B$. In fact, there is no absolute reference for weight values of R, G and B.

B. Noise Removal

1) Imperfection in the scanner intensity of light

2) Scratches or dirt on the camera or scanner lens etc.

introduces noises in the scanned signature images. A filtering function is used to remove the noises in the image. It is required to eliminate single white pixels on black background and single black pixels on white back ground. In order to eliminate the noise we apply a 3 x 3 mask to the image with a simple decision rule: if the number of the 8-neighbors of a pixel that have the same color with the central pixel is less than two, then reverse the color of the central Pixel. The Gaussian filter is used for the noise removal. Since Gaussian function is symmetric, smoothing is performed equally in all directions, and the edges in an image will not be biased in particular direction.

C. Edge Detection

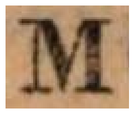
This is the important step in our method .In this stage character on the cards can be highlighted.
For this purpose we will use canny edge detection algorithm [6] which is given below.
The Canny Edge Detection Algorithm:

Canny technique is very important method to find edges by isolating noise from the image before find edges of image, without affecting the features of the edges in the image and then applying the tendency to find the edges and the critical value for threshold. The algorithmic steps for canny edge detection technique are follows:

1. Convolve image f(r, c) with a Gaussian function to get smooth image f^(r, c).f^(r, c)=f(r,c)*G(r,c,6)
2. Apply first difference gradient operator to compute edge strength then edge magnitude and direction are obtain as before.
3. Apply non-maximal or critical suppression to the gradient magnitude.
4. Apply threshold to the non-maximal suppression image.

D. Binarization

To extract the text from the background we need to use another classification. One of these classes represents the text (which is usually represented by black color on the final segmentation map), and the other one represents the background and interfering patterns and is shown as white on the segmentation output. The weak parts of the strokes and text regions with high degree of intensity variations are connected together an to strong portions of the strokes. Therefore, they can easily be differentiated from the complex background and other unwanted information. It can be seen that the true text is accurately segmented, and, despite presence of many weak parts of the strokes, the extracted text is continuous and all connections and loops are preserved. Following figure shows the input image and image after Binarization.



(a) Shows the original image.

(b) Shows the output of Binarized process.

E. Segmentation

As the characters on the card don't appear alone, we have to separate characters into text string groups. Such groups can be observed by the similarity of character size as well as the spatial distribution. Based on the images in which noise regions are reduced, we merge remaining regions into several strings, and distribute these strings into different images according to average width of strokes.

Segmentation subdivides an image into its constituent parts or objects. The level to which this subdivision is carried depends on the problem being viewed .In our case we need to segment the object from the background to read the image correctly and identify the text from the image for this
We will use following algorithm for segmentation:

1) The Binarized image is checked for inter line spaces.
2) If inter line spaces are detected then the image is segmented into sets of paragraphs across the interline graph.
3) The line in the paragraph are scanned for horizontal space intersection with respect to background. Histogram of image is used to detect the width of horizontal lines. Then the line are scanned vertical space intersection.

F. Classification using ANN

Artificial neural networks are computational models which have taken their inspiration from the models and theories of the human brain. The most popular neural network is the multilayer feed-forward network where neurons are grouped as layers and connections between neurons in consecutive layers are permitted. The inputs are fed from the input layer and outputs are at the output layer. In our method, we will use multilayer perceptron network (MPL) with back-propagation learning (also called the back-prop model). This model operates in two distinct phases, the first is the recall phase in which the training pattern is presented to the input layer of the network and a corresponding output is recalled at the output layer. The second is the learning phase in which the network adjusts its synaptic weights in order to minimize the error between the recalled pattern and the correct pattern given by a teacher, see [7] for more details about back-prop model.

For the training of Neural Network we require input feature of the image. This feature extraction process is discussed shortly below.

G. Features Extraction

In this phase, the *texture* features such as *homogeneity* and *contrast* can be obtained from every 50 x 50 block of the processed segmented image *g(x, y)* at 00,450,900, and 1350orientations. Totally 8 features can extracted from every block and are stored into a feature vector *Xi* (Subscript "i" corresponds to i'th block). The feature vector *Xi* also records block coordinates which corresponds to minimum and maximum row and column numbers of the block. Feature vectors of all *N* blocks are combined to form a feature matrix *D* as depicted in equation 5. The feature vector is described in equation below.

$D = [\ X1, X2, X3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots XN]$ T

$Xi = [rmin, rmax, cmin, cmax, fj,, j=1,8]$;

Where;

rmin, rmax, cmin, cmax corresponds to coordinates of ith block in terms of minimum and maximum row and column numbers. f1 and f2 corresponds to mogeneity and contrast at 0 degree orientation. f3 and f4 corresponds to homogeneity and contrast at 45 degree orientation. f5 and f6 corresponds to homogeneity and contrast at 90 degree orientation. f7 and f8 orresponds to homogeneity and contrast at 135 degree rientation.

CONCLUSION AND FUTURE WORKS

The method which is discussed in this paper is computationally simple but promising. Although images used here are grey-scale visiting card images, for color or other types of images, we just need to make some simple modification on the method to work on those images because the main elements processed in this approach are just the edges and edge of color images can be calculate very easily. Although we only discuss visiting card images here, other images like book covers, poster or even images extracted from video will also have similar difficulties so that our method will give a promising result for them. For the means of portability of the visiting card scanner, we use simple edge detection method which can be replaced by a more accurate method like canny edge detector. Further works can also be done in future to improve edge classification by using more parameters. A future focus can be on how to detect light contrast text without being confused by background textures.

REFERENCES

[1] L. Graham, Y. Chen, T. Kalyan, J. H. N. Tan, and M. Li.Comparison of Some *Thresholding Algorithms for TextlBackground* SEGMENTATION IN Difficult Document Images. ICDAR, 2003, Vol 2, pp. 859-865.

[2] S. Wu and A. Amin. *Automatic Thresholding of Gray-level Using Multi-stage Approach*. ICDAR, 2003, pp. 493-497.

[3] H. Negishi, J. Kato, H. Hase, and T. Watanabe. *Character Extraction from Noisy Background for an Automatic Reference System*. ICDAR1999 pp. 143-146.

[4] P. Matti and 0. Okun. *Edge-Based Method for Text Detection from Complex Document Images*. ICDAR, 2001, pp. 286-291.

[5] C. Strouthopoulos, N. Papamarkos, A. Atsalakis, and C. Chamzas. *Text Identification in Color Documents*. ICPA, 2003, Vol 2, pp. 702-705.

[6] John Canny. *A computational approach to edge detection. Pattern Analysis and Machine Intelligence*, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.

[7] B. Soucek and M. Soucek," *Neural and cmzcsivelypallel cxmptms*", Wm Interscience, 1988.