OPTICAL CHARACTER RECOGNITION FOR HANDWRITTEN TEXT

M.Krishnapriya
Department of Computer Science &Engineering
Sri Ramakrishna Engineering College
Coimbatore, Tamil Nadu, India

B.Manigandan, R.Manikandan, B.Chandrasekar
Department of Computer Science &Engineering
Sri Ramakrishna Engineering College
Coimbatore, Tamil Nadu, India

Abstract: the optical character recognition is a windows application. This application combines the functionality of optical character recognition and speech synthesizer. The objective is to develop user friendly application which performs image to text. The OCR takes image as the input, gets text from that image and then converts it into speech. This system can be useful in various applications like banking, legal industry, other industries, and home and office automation.it mainly designed for people who are unable to read any type of text documents.in this paper, the character recognition method is presented by using OCR technology.

Keywords : Optical Character Recognition (OCR), Microsoft Document Image (MODI).

1. INTRODUCTION

OCR is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files or image captured by a digital camera onto editable and searchable data. This technology allows to automatically recognize characters through an optical mechanism.in OCR processing, the scanned image or bitmap is analyzed for light and dark areas in order to identify each alphabetic letter or numeric digit.when character is recognized, it is converted into an ASCII code.

Optical Character Recognition (OCR) is a technique used to convert handwriting into editable text. Handwriting is scanned via a scanner into an image file. OCR software scans the image and matches characters with those in its database and converts it back to searchable and editable text. OCR handwriting is used by archivists to convert handwritten documents to digital form. As research into OCR technology gathers momentum, PDAs and cellular phones have now been incorporated with this technology. Research is still underway in ICR or Intelligent Character Recognition technology is also used to recognize handwriting. Hand printed characters can be converted to ASCII text. Accuracy levels on handwritten text is said to be higher with ICR than by OCR. It is the mechanical or electronic conversion of images of typewritten or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statement, receipts, business card, mail, or other documents. It is a common method of digitizing printed texts so that it can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machinetranslation, text-to-speech, key data and textmining. OCR is a field of research in patternrecognition, artificial intelligence and computer vision.

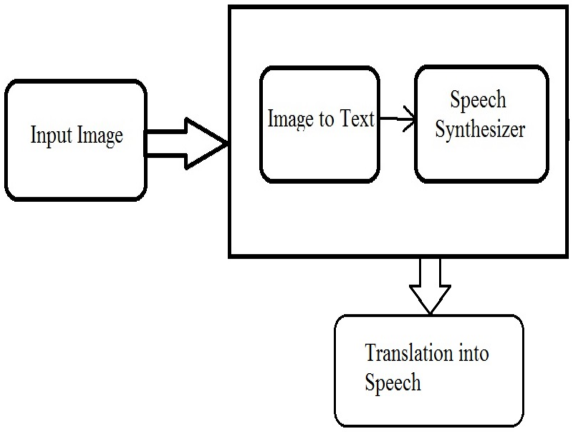

Fig 1.1 Block Diagram of OCR

OCR can recognize both handwritten and printed text. But the performance of OCR is directly dependent on quality of input documents. OCR is designed to process images that consist almost entirely of text, with very little non-text clutter obtain from picture captured by mobile camera. This application is for the windows operating system that combines open-source OCR engine, Tesseract, text recognition OCR engine. Advanced systems that have a high degree of recognition accuracy for most fonts are now common. The goal of Optical Character Recognition (OCR) is to classify optical patterns (often contained in a digital image) corresponding to alphanumeric or other characters. The process of OCR involves several steps including segmentation, feature extraction, and classification. Each of these steps is a field unto itself, and is described briefly here in the context of a Mat lab implementation of OCR.

TESSERACT

Tesseract is Open source OCR engine.tesseract works with independently developed Page Layout Analysis Technology. Hence tesseract accepts input image as a binary image. tesseract can handle both, the traditional- Black on White text and also inverse-White on Black text.Outlines of component are stored on connected Component Analysis. Nesting of outlines is done which gathers the outlines together to form a Blob. Such Blobs are organized into text lines. Text lines are analyzed for fixed pitch and proportional text. Then the lines are broken into words by analysis according to the character spacing. Fixed pitch is chopped in character cells and proportional text is broken into words by definite spaces and fuzzy spaces.

COMPONENT OF OCR

The Fig1.2 given below is illustrates the overall functioning of Optical Character Recognition(OCR).It contains some steps to recognize text. These steps are: scanning,segmentation,preprocessing.Here the input image to OCR is any hand written or printed text like books, general, magazines, newspapers ect.Such input is given to OCR.Firstly it is scanned using camera or pc scanner.It means it digitizes the analog document.

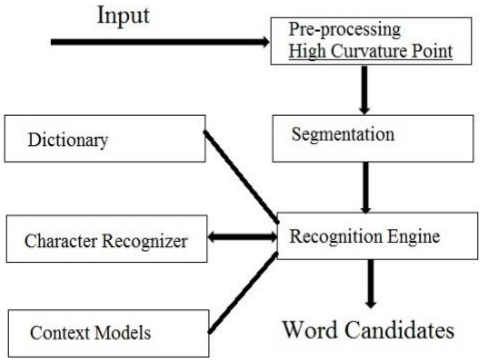

Fig 1.2 OCR Preprocessing

SCANNING

This application i.e.OCR system uses mobile camera or pc scanner.Camera captures image of document.This is nothing but the process of scanning.In short we can say that scanning makes original document as digital image.Generally,original document are made up of the backcoloured text print on the white colored background.Scanning comes with thresholding which makes the digital image as gray scale image.

SEGMENTATION

The process of locating regions of printed or handwritten text is segmentation differs text from figures and graphics. When segmentation is applied to text, it isolates characters or words. The mostly occurred problem in segmentation is: it causes confusion between text and graphics in case of joined and split characters. Usually, splits and joints in the characters causes due to scanning. If document is dark photocopy or if it scanned at low threshold, joints in characters will occur. And splits in characters will occur if document is light photocopy or scanned at high threshold. OCR system also gets confused during segmentation when characters are connected graphics.

PREPROCESSING

As we seen above, some noise may occur during scanning process. This results in poor recognition of characters. This usually occurred problem is overcome by preprocessing. It consists of smoothing and normalization. In smoothing, certain rules are applied to thecontents of image with the help of filling and thinning techniques. Normalization is responsible to handle uniform size, slant and rotation of characters.

FEATURE EXTRACTION

It extracts the features of symbols. Features are the characteristics. In this, symbols are characterized and unimportant attributes are left out. The feature extraction technique does not match concrete character patterns, but rather makes note of abstract features present in a character such as intersections, open spaces, lines, etc. Feature extraction is concerned with the representation of the symbols. The character image is mapped to a higher level by extracting special characteristics of the image in the feature extraction phase.

RECOGNITION

OCR system works with Tesseract algorithm which recognizes characters. tesseract identifies characters in foreground pixels, called as blobs, and then it finds lines. word by word recognition of characters is done throughout the lines. Recognition involves converting these image to character streams representing letters of recognized words. In short, recognition extracts text from images of documents.

METHODOLOGIES

The proposed system is an enhanced version of the existing system. The proposed system can be achieved by following two methodologies are

1 .Determining character lines
2. Detecting individual symbols
3. Symbol image matrix mapping.

1. Determining character lines

Enumeration of character lines in a character image ('page') is essential in eliminating the bounds within which the detection can proceed. Thus detecting the next character in an image does not necessarily involve scanning the whole image all over again.

Algorithm

```
1. Start at the first x and first y pixel of the image pixel (0, 0), Set number of lines to 0
2. Scan up to the width of the image on the same y-component of the image
        A. if a black pixel is detected register y as top of the first line
        B. if not continue to the next pixel
        C. if no black pixel found up to the width increment y and reset x to scan the next
        Horizontal line
3. Start at the top of the line found and first x-component pixel (0, line_top)
4. Scan up to the width of the image on the same y-component of the image
        A. if no black pixel is detected register y-1 as bottom of the first line. Increment number
        Of lines
        B. if a black pixel is detected increment y and reset x to scan the next horizontal line
```

5. Start below the bottom of the last line found and repeat steps 1-4 to detect subsequent lines

6. If bottom of image (image height) is reached stop.

2. Detecting Individual symbols

Detection of individual symbols involves scanning character lines for orthogonally separable images composed of black pixels.

Algorithm

```
1. Start at the first character line top and first x-component
2. Scan up to image width on the same y-component
        A. if black pixel is detected register y as top of the first line
        B. if not continue to the next pixel
3. Start at the top of the character found and first x-component, pixel (0, character_top)
4. Scan up to the line bottom on the same x-component
        A. if black pixel found register x as the left of the symbol
        B. if not continue to the next pixel
        C. if no black pixels are found increment x and reset y to scan the next vertical line
5. Start at the left of the symbol found and top of the current line, pixel (character_left, line_top)
6. Scan up to the width of the image on the same x-component
        A. if no black characters are found register x-1 as right of the symbol
        B. if a black pixel is found increment x and reset y to scan the next vertical line
7. Start at the bottom of the current line and left of the symbol, pixel(character_left,line_bottom)
8. Scan up to the right of the character on the same y-component
        A. if a black pixel is found register y as the bottom of the character
        B. if no black pixels are found decrement y and reset x to scan the next vertical line
```

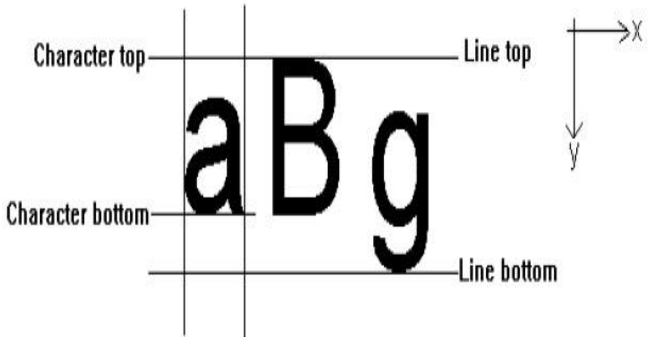

Fig: 1.3 Character boundary detection

3. Symbol Image Matrix Mapping

The next step is to map the symbol image into a corresponding two dimensional binary matrix. An important issue to consider here will be deciding the size of the matrix. If all the pixels of the symbol are mapped into the matrix, one would definitely be able to acquire all the distinguishing pixel features of the symbol and minimize overlap with other symbols. However this strategy would imply maintaining and processing a very large matrix (up to 1500 elements for a 100x150 pixel image). Hence a reasonable tradeoff is needed in order to minimize processing time which will not significantly affect the separability of the patterns. The project employed a sampling strategy which would map the symbol image into a 10x15 binary matrix with only 150 elements. Since the height and width of individual images vary, an adaptive sampling algorithm was implemented.

Algorithm

```
A. For the width (initially 20 elements wide)
1. Map the first (0, y) and last (width, y) pixel components directly to the first (0, y) and last
   (20, y) elements of the matrix
2. Map the middle pixel component (width/2,y) to the 10th matrix element
3. Subdivide further divisions and map accordingly to the matrix
```

B. For the height (initially 30 elements high).

1. Map the first x,(0) and last (x,height) pixel components directly to the first (x,0) and last (x,30) elements of the matrix.
2. Map the middle pixel component (x,height/2) to the 15th matrix element.
3. Subdivide further divisions and map accordingly to the matrix.

C. Further reduce the matrix to 10x15 by sampling by a factor of both the width and height

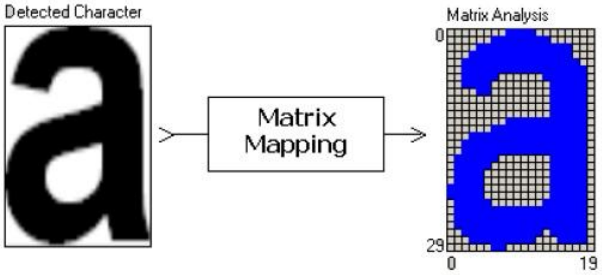

Fig: 1.4Binary matrix

ASPRICE OCR

Asprice OCR is a high performance OCR engine which offers APIs for Java, VB.NET,CSharp.NET, C, C++, Delphi, VB6.0, C (asprise.com). Asprice is a very useful tool as itenables you to equip your applications with OCR ability easily.Computer systems equipped with OCR system can improve the speed of input operationand decrease human errors. Recognition of printed characters is itself a challengingproblem since there is a variation of the same character due to change of font orintroduction different type of noises. Therefore, a good character recognition approachmust eliminate the noise after reading binary image data, smooth the image for betterrecognition, extract features efficiently, train the system and classify patterns.

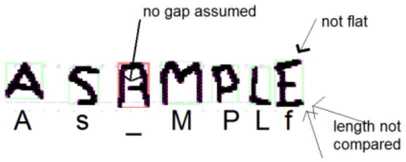

Fig 1.5 Recognize Pixel Paten

CONCLUSION

Thus in this paper the conversion of handwritten into normal editable form is achieved. In this paper the different handwritten styles is scanned from the image, identifies the font style and then it is converted into an editable text file. This system used an OCR engine for the conversion methods. It processing the image scanning, extraction, and then covert the handwritten image to editable text.

REFERENCES

[1] Chucai Yi, *Student Member, IEEE*, YingliTian, *Senior Member, IEEE*, and Aries Arditi ”Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons”, *IEEE/ASME Transactions on mechatronics, vol. 19, no. 3, june 2014.*

[2] RavinaMithe, SupriyaIndalkar, NilamDivekar “Optical Character Recognition”,*International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-2, Issue-1, March 2013*

[3] PingpingXiu, Member, IEEE, and Henry S. Baird, Fellow, IEEE ”Whole-Book Recognition”, *IEEE Transactions on pattern analysis and machine intelligence, vol. 34, no. 12, december 2012.*

[4] C. Yi and Y. Tian, "Text string detection from natural scenes by structure based artition and grouping," *IEEE Trans. Image Process.*, vol. 20, no. 9,pp. 2594–2605, Sep. 2011.

[5] D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," *IEEE Trans. Syst., Man, Cybern.*, vol. 40, no. 1, pp. 25–35, Jan. 2010.