AUTOMATED FORM READING

CHU KIM CHING

M. PHIL.

THE HONG KONG
POLYTECHNIC UNIVERSITY
2000

Abstract

Forms are used extensively to collect and distribute data. The main task of an automated form reader is to locate the data filled in forms and to encode the content into appropriate symbolic descriptions. In this thesis, we aim to develop efficient algorithms for an automated form reading system. The key problems tackled in this thesis are image preprocessing, script determination, fast keywords matching, and printed character recognition.

Preprocessing of digital images is a very important step in document analysis. We discuss in this thesis a combined intensity histogram and a local contrast feature for image binarization, horizontal Run Length Smoothing Algorithm (RLSA) followed by 8-neightbouring connection method for page segmentation, the use of simple criteria for text and line extraction, and fast skew estimation and correction using the extracted lines with a backup of an interline cross-correlation method for those forms without lines. Our approach for skew estimation is efficient and effective for those forms containing a lot of lines. The skewed angle can be detected with an error of smaller than 1°.

It is very common that the documents contain more than one script. In this thesis, a robust script determination approach is proposed which can cope with different fonts, sizes, styles and darkness of text in document images. Two neural networks are employed. The first neural network is trained to derive a set of 15 masks which are used for extracting 15 features. The coefficients of masks are then quantized for reduced computational complexity. The second neural network is trained with 15 extracted features to perform the script separation. Experimental results show that 97% of the image can be correctly classified.

A Dynamic Recognition Neural Network (DRNN) is proposed in this thesis to perform fast keywords matching. Different sets of features are used to deal with different scripts. For English, projection profiles (x and y) are used while for Chinese, contour features are utilized. Testing on 29 name cards shows that a 90% correct matching rate can be achieved.

An algorithm based on the vertical projection and a peak-to-valley function is adopted for segmenting characters. By applying the algorithm on form images with 100 *dpi* scanning resolution, about 86% of the characters can be correctly segmented. A neural network is then employed to classify the segmented characters into 50 groups. Both intensity features and structure-based features extracted from the skeleton image were utilized. An accuracy of 85% to 87.5% can be achieved when testing on the images with 100 *dpi* scanning resolution and higher accuracy of 94% to 96.6% can be achieved if the scanning resolution is 150 *dpi*.

Acknowledgements

I would like to express sincere gratitude to various bodies from The Hong Kong Polytechnic University, where I have opportunity to work with. My major debt is to my Chief Supervisor Dr. Zheru Chi, for his warmness help and his professional advises concerning my study.

I would also like to thank Prof. W. C. Siu, my second supervisor and the Head of the Department of Electronic and Information Engineering. Without his kind support, this study could never reach its present level.

I am greatly indebted to many people with whom I had numerous discussions and who supplied me with various materials. Particular thanks go to Dr. L. P. Zhang, Mr. Z. K. Lu, Mr. J. Kong, and Mr. Z. Y. Wang. Their expert knowledge and friendly encouragement have helped me in resolving many difficult problems in my study.

I have also to thank all my colleagues and researchers of the DSP Research Group for their continuous supports and inspirations. Thanks also to the other members of Department of Electronic and Information Engineering for their friendship and help.

It is my pleasure to acknowledge the Research and Postgraduate Studies Office of the Hong Kong Polytechnic University for providing financial support to me over the last few years.

Finally, I must express my heartfelt gratitude to my grandparents, parents, uncles, brothers, Miss Cinderella Tsoi, Dr. Richard Hsung, Mr. P. Y. Ng, Mr. C. F.

Cheng, Mr. Eric Chu, Mr. Kelvin Tsang, Mr. Andrew Liu, Mr. Bobby Mak, Mr. W. H. Pin, and Mr. Derek Ng for their endless support and encouragement. Without them, this study would not have the chance to be completed.

Statement of Originality

The work described in this thesis was carried out at the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University, between September 1996 and August 1999, under the supervision of Dr. Zheru Chi and Professor Wan-Chi Siu.

The thesis consists of six chapters and one appendix. The work described in this thesis was originated by the author except where acknowledged and referenced, or where the results are widely known. The following is the statement of original contributions.

1. A combined histogram and local contrast features for image binarization was presented by the author. (Chapter 2, Section 2.2)
2. Combining 8-neightbor connection method with horizontal Run Length Smoothing Algorithm (RLSA) for page segmentation is the work of the author. Overlapped blocks can be successfully separated by this method. (Chapter 2, Section 2.3)
3. Combining the number of vertical transitions with the aspect ratio in binary image blocks to determine straight lines in form images is the work of the author. (Chapter 2, Section 2.4)
4. Approximating the weight coefficients in 15 neural network derived masks to 17 pre-defined values for reduced computational complexity for script

determination was joined work of the author and Dr. Zheru Chi. (Chapter 3, Section 3.2)

5. Using a smaller neural network for script determination of Chinese and English textural images was modified by the author and Dr. Zheru Chi from an approach presented by Jain and Yu in 1996 [7]. The implementation work was done by the author. (Chapter 3, Section 3.3)

6. Using a Dynamic Recognition Neural Network (DRNN) for speech recognition was proposed by Zhang *et al* [57]. The application of a DRNN to fast keywords matching in document image processing is a joined work of the author, Dr. Lipeng Zhang and Dr. Zheru Chi. (Chapter 4)

7. A neural network character classifier with combined intensity and skeleton features as the input was implemented by the author. (Chapter 5)

8. Implementation of a prototypic form reading system is the work of the author.

Publications

Journal Paper

1. K. C. Chu, Z. Chi and W. C. Siu, "A Neural Network Approach for Language Separation of Document Images," *Chinese Journal of Electronics*, **Vol.** 7, No. 4, pp. 381-386, October 1998, China.

Conference Papers

1. K. C. Chu and Z. Chi, "An automatic form reading system," *Proceedings of the 2nd IEEE International Conference on Intelligent Processing Systems (ICIPS'98)*, pp. 497-501, August 4-7, 1998, Gold Coast, Queensland, Australia.
2. K. C. Chu and Z. Chi, "A neural network model for language determination of textual document images," *Proceedings of 1998 International Conference on Neural Networks and Brain*, pp. 597-600, October 27-30, 1998, Beijing, China.
3. L. Zhang, Z. Chi and K. C. Chu, "A dynamic recognition neural network for keywords matching in document processing," *China Fourteen National Conference on Circuits and Systems*, pp. 382-385, Fuzhou, China, April 7-10, 1998. (Invited Paper)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of Automated Form Reading

Automated document analysis and understanding is a very active research area because of its wide range of applications such as form and check reading and document image retrieval [1,2,3]. Many investigations have been carried out in the three main aspects of document image processing: page segmentation [4,5,6], document image analysis [1,7], and character recognition [8,9].

There are a great number of different types of document images with different styles and purposes. It is difficult, if not impossible, to develop algorithms that can handle all types of document images. One solution is that we only work on a specific type of documents such as forms.

Forms are used extensively to collect or distribute data in. They represent a vast majority of the paperwork need to conduct business. Hence, forms processing [10,11,12,13] is becoming very important in practice, since those documents have typically a simple structure when compared with general documents such as articles that may contain pictures, figures, mathematical symbols, and complex background.

Moreover, the automation of the form reading process is an objective of relevant interest, since form processing is in fact an essential operation in many business organizations.

In this thesis, we focus on developing efficient algorithms for an automated form reading system. The system would be able to learn the layouts of a variety of forms, to locate and extract data correctly and to recognize the data such as Chinese/English characters, digits or symbols, so that the useful data can be extracted from the forms and stored in appropriately according to the needs of the user.

1.2 Previous work on Machine Form Reading

The overall objective of a form reading system is to obtain a symbolic representation of the data (either printed or handwritten) from a given form. The typical operations involved in these systems are document image acquisition and preprocessing, layout analysis, and data interpretation [11,12].

The preprocessing usually involves binarization followed by form registration. During the layout analysis, important regions that contain the information are located in the form. Finally the data interpretation step takes place on regions extracted previously by either handwritten or printed character recognition.

Form recognition is usually based on the assumption that the information fields are in fixed positions. Another approaches [14] for information fields extraction are by the detection of lines. Lines can be described by orientation, thickness, and position in the form. They can be used as a reference for detecting information fields. However, a more flexible way to characterize information fields is to use the corresponding instruction fields to describe the positions and the content of the information fields. In 1991, Lam and Srihari [15] proposed a system to locate related instruction and information fields by recognizing the text of the instruction fields. Fujisawa *et al* suggested that only fields that are inside rectangles (referred to as frames) should be taken into account [7]. The frames were classified into two classes: label frames and data frames, which contain instruction fields and information fields respectively. They used the spatial relationship between the two rectangles to link up a label frame and the corresponding data frame.

Usually, ones should consider whether or not the form is known in advance before performing form registration and layout analysis as in different cases, different operations are needed [11]. If the form is known in advance, that is, if its layout is previously defined, then the data extraction can be based on a top-down, model-driven process [16]. In this case, the form registration will be the most important step as data will directly be extracted from the pre-defined positions once the registration has taken place. Failure in the form registration step means that all the data will be incorrectly extracted. In the case of unknown forms, a possible approach is to classify documents in a fixed number of classes. Then use the correspondent model of each class to carry out the layout analysis [17,18].

1.3 Previous Work on Document Image Processing

Many algorithms have been developed for general document image processing such as binarization [19,20] , page segmentation [4,21,22], skew detection and correction [11,23], text and lines extractions [24,25,26], character segmentation [8,27,28], script determination [29,30] and character recognition [31,32,33].

The popular tool for binarization is image thresholding, which transforms a gray scale image into a binary image in which, foreground and background data are separated. The quality of the binarization step is critical for subsequent analysis. If poorly binarized images are used, document understanding would be difficult or even impossible. Thresholding techniques can be categorized into two classes: global and local. Global thresholding algorithms use a single threshold that will be applied on all the pixels in the image, while local thresholding algorithms compute an individual threshold for each pixel based on a neighborhood of the pixel.

Different page segmentation techniques have been proposed in the many literatures. In the early approaches of page segmentation, the assumption was made that an input document image consists of right rectangular blocks. And they were typically applied to binary images. These methods can be classified as either top-down or bottom-up approaches. In a top-down strategy, a page is first split into major regions, and major regions into subregions, and so on. On the other hand, in a bottom-up strategy, connected components are merged into small regions based on local evidence, and the small regions are then successively merged into larger regions. Recent hybrid approaches to page segmentation have utilized both local and global strategies to cope with complicated page segmentation problems.

Several methods have been developed by many researchers for line and skew angle detection. The Hough transform (HT) is a well-known method for line detection from raster images. Its development and extension can be found in recent survey articles [34,35,36]. Besides that, Ciardiello *et al* have suggested a method using the projection histogram [37]. In this method, a sample region with the maximum average density of black pixels per row is rotated by prespecified angles. The horizontal projection histogram of the region is evaluated for each angle. The skew angle can be found by choosing a rotated angle which maximize the mean square deviation of the histogram.

For text extraction, many techniques have been developed through the years that can be divided into three main categories: top-down, bottom-up and hybrid techniques. Bottom-up techniques [26] are usually based on connected components analysis: they usually connect the neighboring pixels to form symbols, words, text lines and so on by continuously increasing the thresholds. In this approach, no assumption is made on font style, font size nor about page layout. However, the approach is computationally very intensive and the performance relies very much on the use of appropriate thresholds. Top-down techniques use the assumptions about the layout of the page in order to segment it [38,39,40]. They are usually faster than bottom-up techniques and they are efficient for special purposes. But they are not suitable for more general needs. For those proposed methods which can not be fit into any of these two categories, they will be categorized as hybrid.

The main factors that affect the complexity of character segmentation in machine-printed text are the wide variety of fonts, rapidly expanding text styles, and image characteristics. Moreover, the thresholding processes in the binarization step, as well as the thinning algorithms, can cause broken, touching and merged characters which make the character segmentation more difficult.

For the text with proportional pitch, they can be divided into broken characters and touching characters. There are two methods for segmenting broken characters. One is to employ a merging process based on the estimated character width and intervals, and the other is to combine character components based on recognition results [41]. On the other hand, the touching characters are often segmented using structural analysis. Lu *et al* proposed a character segmentation algorithm based on the analysis of neighboring components [25]. The algorithm consists of two steps: computing the connected components and analyzing the structure of neighboring components.

Merged touching characters are components of connected multiple characters. Segmentation of touching characters has been the most difficult problem in character segmentation. The segmentation techniques have to determine whether a segment contains multiple characters or not, and find the suitable break locations if necessary. The techniques for segmenting touching characters can be divided into two categories, featured-based or recognition-based. Some of the touching characters can only be segmented by recognizing the individual components while some others can be recognized only after they have been isolated. Therefore, techniques in both categories are equally important. The vertical projection is widely used technique for character segmentation [41]. To deal with the touching characters in the first category, the algorithms often transfer the vertical projection to functions which provide more information for finding the break points. Decision can be made based on these function values. The commonly used functions include characteristics of the word or line images, such as width, height of the word and candidate segment, aspect ratio, foreground pixel density, contour analysis, etc. On the other hand, the results of character segmentation in the second category usually based on the recognition process.

Many techniques have been developed for identifying of the script when the input image contains characters. In 1990, Spitz proposed a method of classifying individual text lines as being either English or Japanese in documents [42]. The method is based on the distribution of an index of optical density. For Japanese text lines, they may contain Kanji which tend to be complex, and therefore optically dense. However, they may also contain Kana which tend to be simple and therefore optically light. On the other hand, English characters have a more consistent optical density. Therefore, this can be used to identify the scripts. By using filtered pixel projection profiles, Wood *et al* proposed a technique that could identify five scripts based on the response to tuning the parameters of the filter [30]. In 1997, Hochberg *et al* proposed a technique for identifying 13 scripts, where single connected components may span the entire horizontal extent of words [29]. Their algorithm has an advantage that is able to accept a new script without tuning of the system. And it is based upon identifying frequently occurring connected component templates that have been scale-normalized.

Character recognition can be performed separately or accompany with the process of character segmentation. One of the most popular approaches is the feature matching method. In general, the feature matching technique needs to extract the complete features of a character pattern to order to achieve the correct recognition. Besides, the syntactic approach is also an active research area in character recognition. A syntactic method is capable of using primitives to describe local details and production rules to describe the global structure. The outer boundary and the skeleton of a character image are two popular forms of images used for character recognition in a syntactic approach. In most cases, the statistical information was incorporated into a string matching process in order to improve the reliability of matching and recognition. Lee proposed a methodology for character segmentation and recognition that makes use of the characteristics of gray-scale images [8]. In the proposed methodology, the character segmentation regions are determined by using projection profiles and topographic features extracted from the gray-scale images. Then a nonlinear character segmentation path in each character segmentation region is found by using multi-stage graph search algorithm. Finally, in order to confirm the nonlinear character segmentation paths and recognition results, recognition-based segmentation method is adopted. Artificial Neural Networks (ANN) for pattern recognition have attracted more attention in the past decade [43]. The advantages of the ANN approaches are their massive parallelism, high noise tolerance, and adaptability. Many ANN approaches for character recognition were proposed throughout the years and satisfactory results were reported [33,44,45].

1.4 Our Work

In this thesis, efficient algorithms for an automated form reading system are developed. The system supports functions including segmenting forms, locating and extracting data reliably and recognizing the data, such as English characters, digits or symbols. By using the system, the useful data can be extracted from forms and stored appropriately according to the needs of the user.

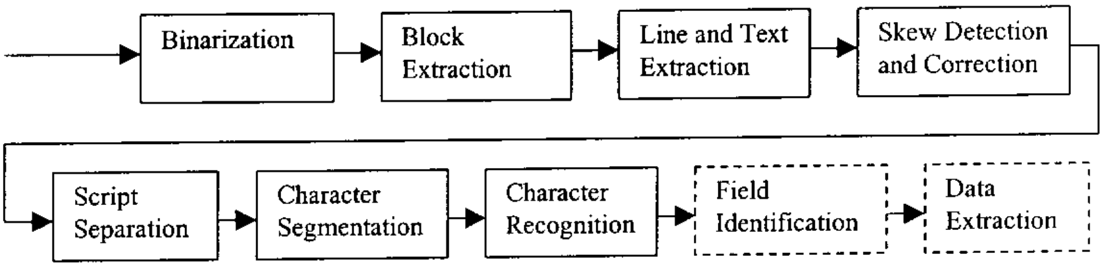


Fig. 1.1 Block diagram of the automatic form reading system

Figure 1.1 shows the block diagram of the automatic form reading system that we developed. A form is digitized and inputted as gray-scale image to a computer by using a scanner. Then it goes through a preprocessing step in which the image is binarized using both the intensity histogram and a measure of local contrast. A Run Length Smoothing Algorithm (RTSA) and 8-neighbourhood connection method is then used to extract blocks that contain useful information. According to the features of the blocks, text and lines are detected and extracted. Based on the extracted lines, the skew angle of the image is determined and the image is then de-skewed. For those blocks that contain text data, script is determined using a neural network approach. The characters in a text block are then segmented using vertical projection and a peak-to-valley function. Finally, character recognition is carried out using a neural network.

Our research focus on high-performance binarization, text block extraction, skew estimation and correction, script determination of Chinese and English textual images, English character segmentation and recognition. We have also made investigation into fast keywords matching using a Dynamic Recognition Neural Network (DRNN).

1.5 Organization of This Thesis

Following this Introduction, the document image processing algorithms used for image preprocessing is discussed in Chapter 2. In chapter 3, a neural network based script determination approach is presented together with experimental results on a number of textual images. A DRNN based algorithm for fast keywords matching is presented in chapter 4. In chapter 5, a printed English character recognition system including both character segmentation and recognition is discussed. Finally, concluding remarks are drawn in Chapter 6.

Chapter 2

Document Image Processing

2.1 Introduction

In form reading, a form-like document is first scanned and stored as a digital image. Several factors that will affect the reading performance include the quality of a document, the distortion introduced during the scanning process, and whether or not an image is skewed. In order to achieve satisfactory reading performance, some preprocessing algorithms should be applied.

Preprocessing of document images has been an active research area for many years [4,11,24,46,47,48,49]. Although there are many existing algorithms for various preprocessing tasks, few have integrated the algorithms into a whole system. A critical problem to combine them into an effective system is to choose the suitable algorithms that are compatible to each other. In this chapter, we will propose several algorithms for form image preprocessing such as binarization, page segmentation, text and lines extraction, and skew detection and correction.

2.2 Binarization of Document Images

It is an important step to separate the foreground information (mainly text) of a document image from the background. There are many existing algorithms for this purpose [19,20,48].

In this thesis, a combination of histogram and a local contrast feature is used to binarise document images. It is found that the method can successfully separate the foreground information from the backgrounds of images as complex as map images in which different intensities are used for the background.

Algorithms

To successfully separate the foreground information from the background, two thresholds are determined and used for making decision of whether a pixel belongs to the foreground or background. One threshold is computed from the histogram of the gray scale image and the other from the histogram of a measure of local contrast.

Let J be a measure of local contrast of a darker pixel against its background. It is defined as

$$J(i,j) = \frac{\max[0, B(i,j) - I(i,j)]\operatorname{sgn}[C(i,j)]}{LA(i,j)} \qquad (2.1)$$

where $I(i,j)$ is the intensity of the pixel in gray-scale and $LA(i,j)$ is defined as

$$LA(i,j) = \frac{\sum_{p=i-3}^{i+3} \sum_{q=j-3}^{j+3} I(p,q)}{49} \qquad (2.2)$$

sgn is the sign operator defined as

$$\mathrm{sgn}(x) = \begin{cases} -1 & : \quad x \le 0 \\ 1 & : \quad x > 0 \end{cases} \tag{2.3}$$

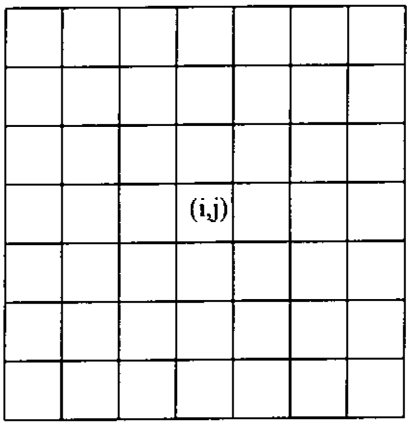


Fig. 2.1: Eight neighboring pixels for calculating *C(i,j)*

The term *C(i,j)*, which is defined in equation (2.4), measures the difference of a pixel intensity and the average intensity of its eight neighboring pixels. Since lines have thickness, to reduce the effect of the foreground pixels within the same line on calculating *C(i,j)*, the nearest neighbor pixels are not included in computing. The eight neighboring pixels for computing *C(i,j)* are shown in Fig. 2.1.

$$\begin{aligned} C(i,j) = \frac{1}{8}[&I(i-3,j)+I(i-2,j)+I(i+2,j)+I(i+3,j)+I(i,j-3) \\ &+I(i,j-2)+I(i,j+2)+I(i,j+3)]-I(i,j) \end{aligned} \tag{2.4}$$

The *B(i,j)* term is a measure of the average intensity of the relative brighter pixels [*C(p,q)* $\le$ *0*] in the 9 × 9 region and is computed by

$$B(i,j) = \frac{1}{N_b} \sum_{\substack{i-4 \le p \le i+4 \\ j-4 \le q \le j+4 \\ C(p,q) \le 0}} I(p,q) \tag{2.5}$$

where N_b is the number of relative brighter pixels.

A foreground pixel usually has a positive value of J, and most background pixels have negative, zero, or small positive values. A threshold is determined by locating a deep valley in the histogram of local contrast J.

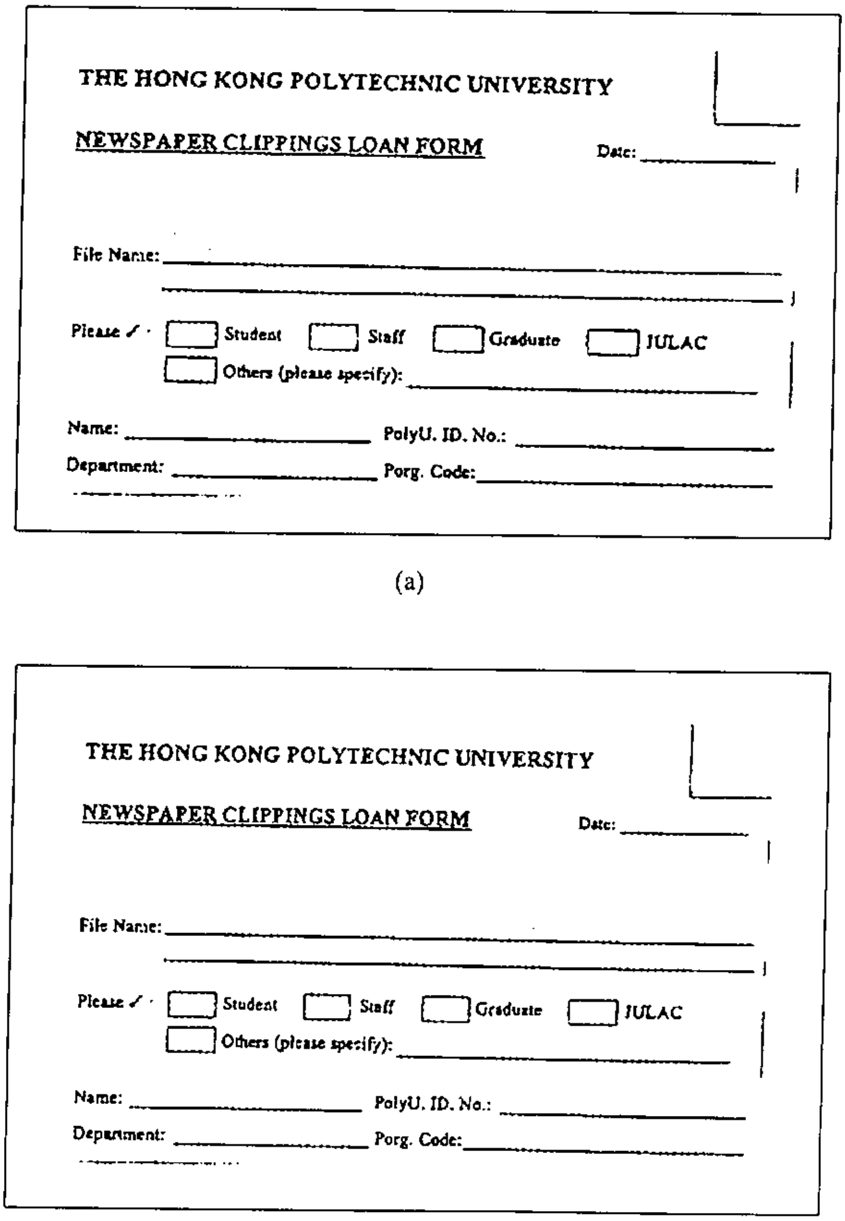
The second feature used is the histogram of pixel intensities of the image. In document images, especially in form images, pixels can be classified into two classes: foreground or background. Therefore, the histogram of the image should contain two peaks close to both ends. An image-adaptive threshold is determined by examining these peaks. Although the absolute locations of the peaks and valleys of histograms are quite different from one image to another, the shape (relative location of peaks and valleys) of the histograms are similar for document images. A threshold can be determined by using the locations of the peaks of the histogram so that it can tolerate the changes of image brightness due to illumination. Let b_p and d_p denote the brightest peak and the darkest peak in the histogram respectively. The threshold (T) is then determined by

$$T = d_p + \lambda(b_p - d_p) \tag{2.6}$$

where λ is a factor with value between 0 to 1. By examining different values of λ on all the form images, it is found that 0.85 is the optimal value for λ.

Experimental Results

Based on the two thresholds discussed above, the foreground information can be separated from the background in the images. Fig. 2.2 shows an example of image binarization.



THE HONG KONG POLYTECHNIC UNIVERSITY

NEWSPAPER CLIPPINGS LOAN FORM Date: ____________

File Name: ______________________________

Please ✓ · □ Student □ Staff □ Graduate □ JULAC

□ Others (please specify): ____________

Name: ____________ PolyU. ID. No.: ____________

Department: ____________ Porg. Code: ____________

(a)

THE HONG KONG POLYTECHNIC UNIVERSITY

NEWSPAPER CLIPPINGS LOAN FORM Date: ____________

File Name: ______________________________

Please ✓ · □ Student □ Staff □ Graduate □ JULAC

□ Others (please specify): ____________

Name: ____________ PolyU. ID. No.: ____________

Department: ____________ Porg. Code: ____________

(b)

Fig. 2.2 The results of foreground and background separation: (a) original image. (b) binarized image.

2.3 Page Segmentation

Page segmentation is a document processing technique used for determining the layout of a page. Before applying classification algorithm, a scanned input image must first be well segmented into different regions. Higher level analysis of the input document image is based on the result of page segmentation. For example, text extraction is based on the features in individual segmented text region but not the whole image. Many algorithms for page segmentation have been developed [4,5,6]. A survey of page segmentation techniques and document image understanding can be found in Haralick [50].

Algorithms

In this thesis, the Run Length Smoothing Algorithm (RLSA) is applied to the binarized image to check the connection between the black pixels. The space between two neighboring black pixels is usually smaller in the same information part, for example, horizontal spacing in a text part, than that in different parts so that a threshold can be set to determine whether the two black pixels are in the same part or not. The RLSA is applied to binary sequence in which black pixels are represented by 1's and white pixels by 0's. The algorithm transforms a binary sequence **x** into an output sequence **y** according to the following rules:

(1) 0's in **x** are changed to 1's in **y** if the number zeros between adjacent 1's is less than a predefined limit *C*.

(2) 1's in **x** are unchanged in **y**.

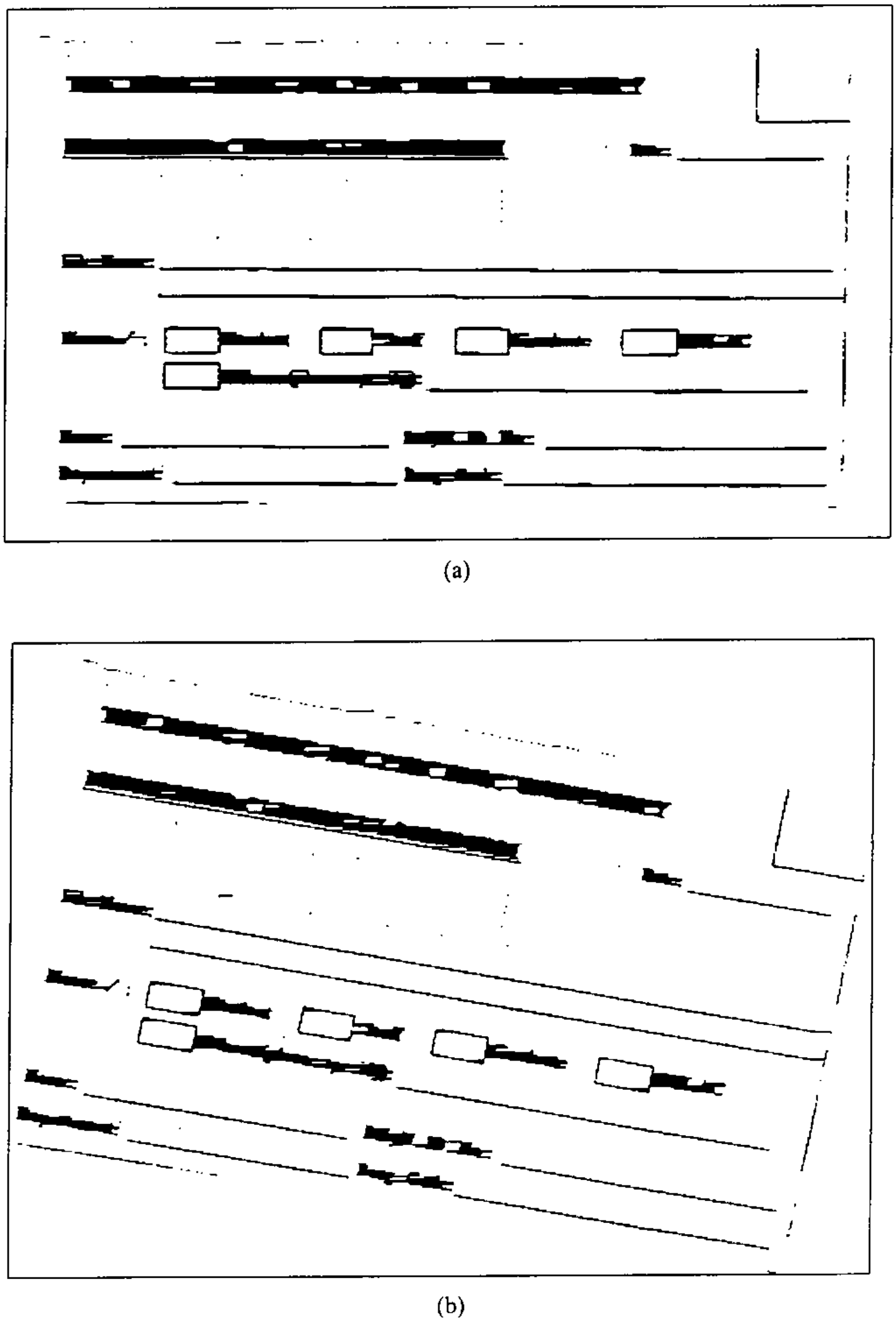
For example, with $C = 5$, the sequence **x** is mapped into **y** as follows:

x: 00011001111000000111011100111111000000000111100000111001101110000 y: 00011111111000001111111111111110000000011111111111111111110000

In form documents, most of the information is arranged in horizontal lines. Therefore, only the horizontal RLSA is applied to identify the connected components. A paragraph in a form will be treated as a number of individual text lines. following by the RLSA, an algorithm is used to check the 8-neighborhood connection between the black pixels. After all the connected pixels are checked within the same part, a block with the smallest size which can just surround that part will be assigned. The algorithm is applied recursively until all the black pixels have been assigned to a part. The advantage of using the 8-neighborhood connection instead of vertical RLSA is the potential of separating text parts from other effects, such as underline and text in boxes, which are most likely to be happened in many form-like documents.

Experimental Results

Figure 2.3 shows the results of applying RLSA on a form image with *C* set to 5.

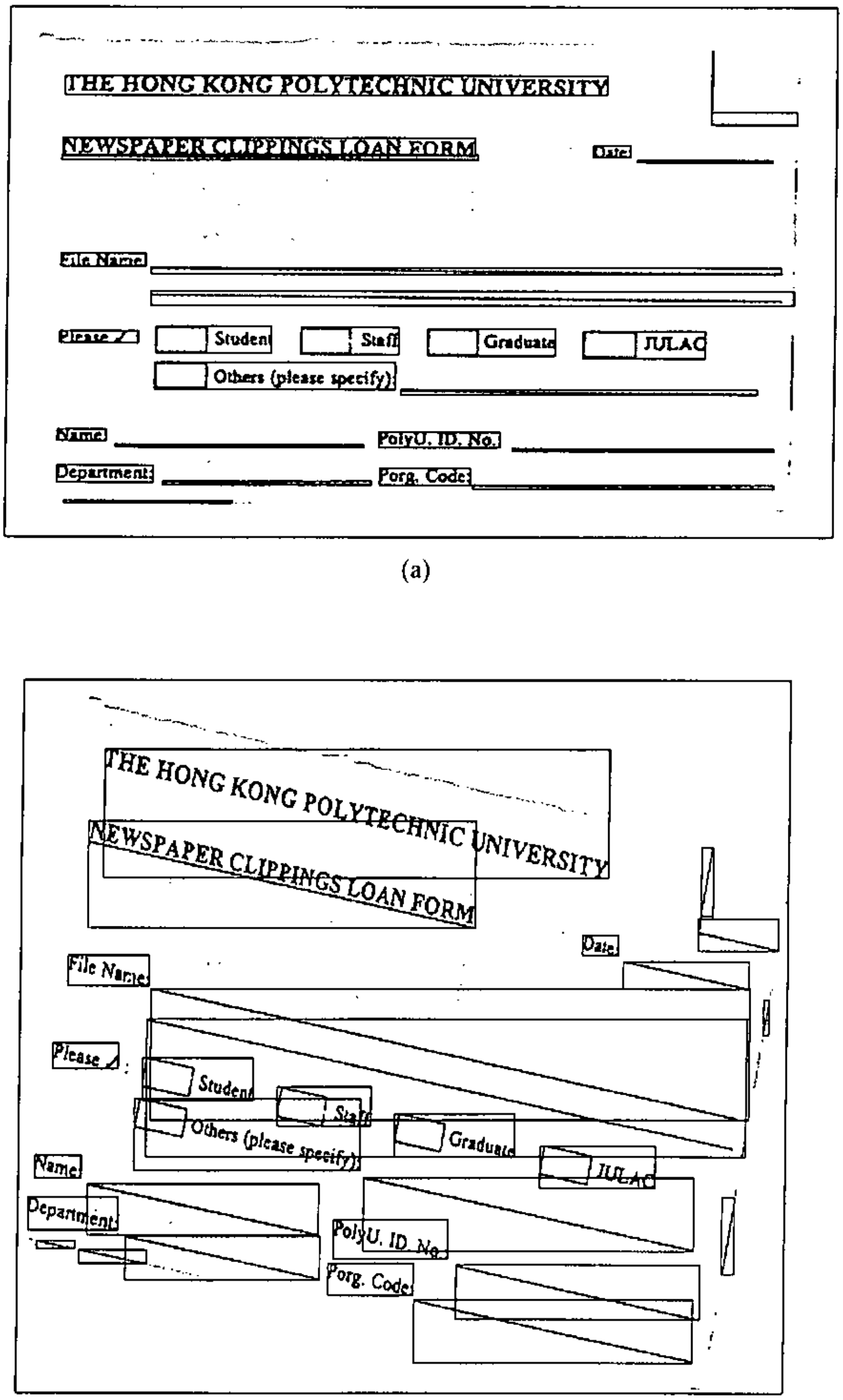


(a)

(b)

Fig. 2.3 Results of applying the Run Length Smoothing Algorithm two form images: (a) an upright form image; (b) a skewed form image.

Page segmentation results of an upright form image and its skewed one are shown in Fig. 2.4.



(a)

(b)

Fig. 2.4 Page segmentation results: (a) an upright form image; (b) a skewed form image.

2.4 Text and Lines Extraction

A key phase of the machine reading of documents is the segmentation of a page into text and other regions, in order to provide appropriate input to the character recognition system [24]. Many techniques for text extraction have been developed through the years. Bottom-up techniques [26] are usually based on connected components analysis while top-down techniques [38,39,40] use a priori assumptions about the layout of the page in order to segment it. On the other hand, lines provide important topological features in form documents and it is very necessary to locate them. Thus, the image blocks extracted by the algorithms discussed in Section 2.3 classified into different categories, text, straight lines, and others, for successive processing. For example, straight lines will be used for skew estimation and correction.

Algorithms

In this thesis, as we have segmented the image, a fast and efficient algorithm has been chose to apply on the blocks extracted in page segmentation for classification in which some simple measurements are taken as the features:

1. The height of a block (*H*)
2. The width of a block (*W*)
3. The mean horizontal length of the black runs in the original data from the block (*R*)
4. The number of vertical cuts for each vertical line in the original data from the block

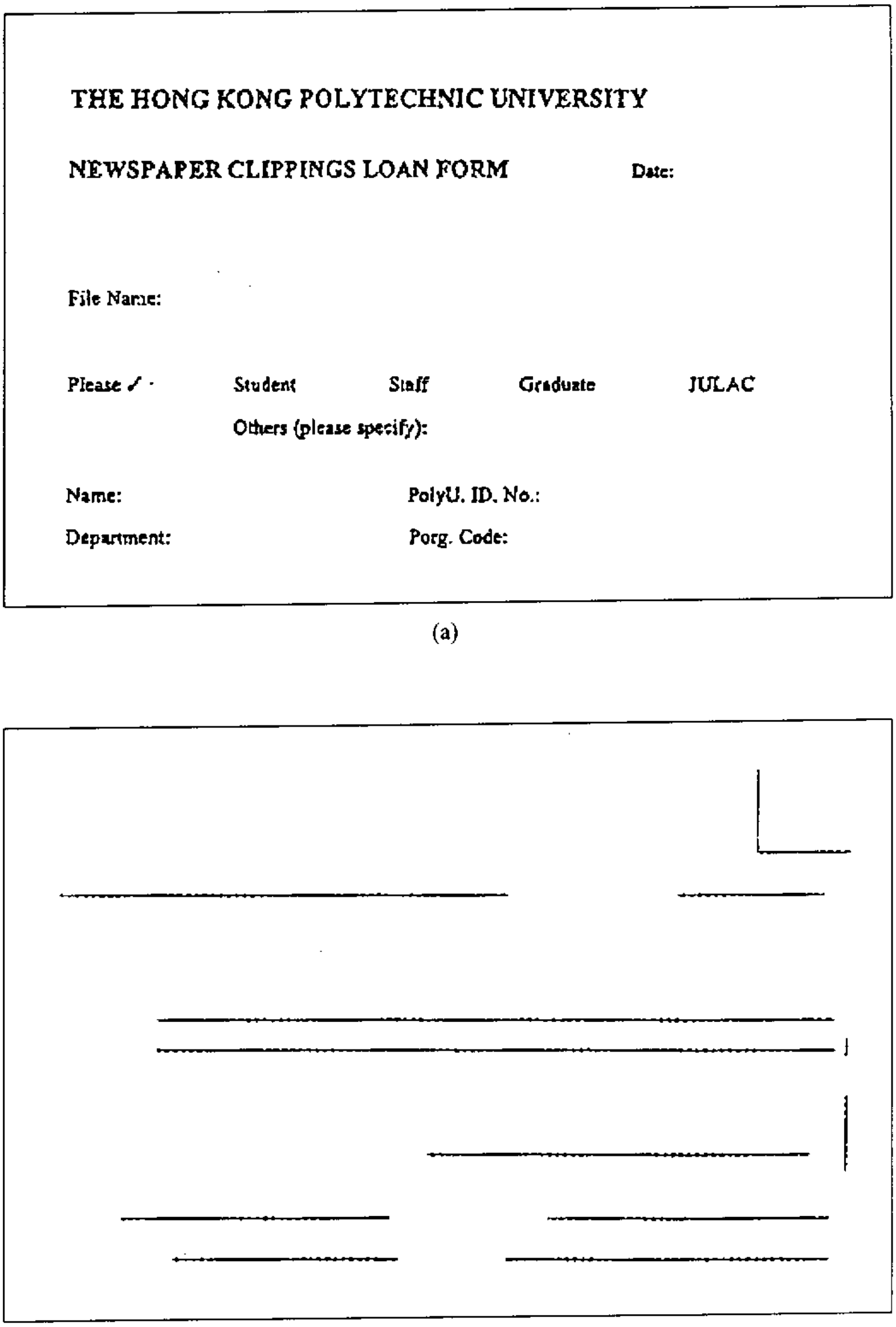
Since text is the predominating data type in a typical form and text lines are basically textured stripes of approximately a constant height H and a mean length of black runs R_m, text blocks tend to be cluster due to these features. For the blocks that contain straight lines, the aspect ratio (A), which is equal to W/H, or its inverse must be much larger than other kinds of information blocks. For the skewed images, the aspect ratio, as well as the horizontal length of black runs, can not determine straight lines correctly. Thus, the number of vertical cuts is used for this purpose. As a result, the blocks can be classified by the following rules:

1. Straight lines if $R > C_R$ and $A > C_A$ or number of vertical cuts equals to 1 for all vertical lines in the block
2. Texts if $R < C_R$ and $A > C_A$ and $H > C_H$
3. Graphics/Images/Noise if $R < C_R$ and $A < C_A$

where C_R, C_A and C_H are pre-set threshold values.

Experimental Results

The values of the parameters are determined based on several training images. With $C_R = 6$, $C_H = 5$ and $C_A = 1.5$. This method has been tested on a number of form images and the results are satisfactory. Examples of text extraction and lines extraction are shown in Fig. 2.5.



THE HONG KONG POLYTECHNIC UNIVERSITY

NEWSPAPER CLIPPINGS LOAN FORM Date:

File Name:

Please ✓ · Student Staff Graduate JULAC

Others (please specify):

Name: PolyU. ID. No.:

Department: Porg. Code:

(a)

(b)

Fig. 2.5 Text and lines extraction: (a) text blocks extraction; (b) lines extraction.

2.5 Skew Estimation and Correction

During the scanning process, a document may be skewed. This may cause problems in document layout analysis and other processing. Several skew estimation methods have been reported in the literature [11,34]. However, the methods involving interpolation of the image and Fourier transform are time consuming for a large image. Hough transform is one of the most popular approaches to detect skew angle of a document image [35,36,51]. This method can be applied to binary images only and the performance depends on the reliability of the character extraction procedure. Moreover, this method is computational intensive.

Algorithms

Lines provide important topological features for form identification and skew information. By using the algorithms discussed in Section 2.4, lines can be located and extracted. The skewed angle of an image can be determined by simply computing the orientations of all the lines and take the majority vote on line orientations.

After determining the skewed angle of the image, we need to correct it by rotation. 2-D rotation of an image rotates the object in a circular path on the 2-D plane. The following equations rotate the image in reference to the origin (the top-left corner of the image):

$$\begin{aligned} X_{new} &= X\cos\theta - Y\sin\theta \\ Y_{new} &= Y\cos\theta + X\sin\theta \end{aligned} \qquad (2.7)$$

where X and Y refer to the horizontal and vertical coordinates of the images. θ is the angle of rotation. If θ is positive, the rotation is in a counterclockwise direction; otherwise, it is in a clockwise direction.

However, there are always some forms that do not contain any line. In these cases, we use the interline cross-correlation method proposed by Yan [11]. In this method, the cross-correlation between a pair of lines in the image with a fixed distance is computed. The correlations for all pairs of lines in the image are accumulated. The vertical shift for which the accumulated cross-correlation takes the maximum is then used for determining the skew angle. The interline cross-correlation method can be used for gray-scale and color images as well binary images.

Assume that the image intensity is represented by a two-dimensional function $i(x,y)$ where $0 \le x \le X$-1 and $0 \le y \le Y$-1, and that the corresponding binary image is $B(x,y)$ with1 representing a bright pixel in the background and 0 a black pixel in the foreground. We consider the following function:

$$R(s) = \sum_{x=0}^{X-d-1} R1(x,s) \tag{2.8}$$

where d is a constant, $-S \le s \le S$, and

$$R1(x,s) = \sum_{y=S}^{Y-S-1} B(x+d, y+s)B(x,y) \tag{2.9}$$

Note that $R1(x,s)$ is the value of the cross-correlation with a shift s between two vertical lines of the image at x and $(x+d)$, respectively, in the horizontal direction and that $R(s)$ is the accumulated cross-correlation function for all pairs of lines with a fixed distance d in the image.

$B(x+d,y+s)B(x,y)$ is 1 only if both (x,y) and $(x+d,y+s)$ are background pixels. Thus $R(s)$ is maximized at a value S_{opt} for which there is a maximal overlap for the white spaces (bright pixels) between text lines for all pairs of vertical lines. The estimation can be very accurate if (X-d-1) is reasonably large. The skew angle α can be determined from

$$\tan\alpha = \frac{S_{opt}}{d} \tag{2.10}$$

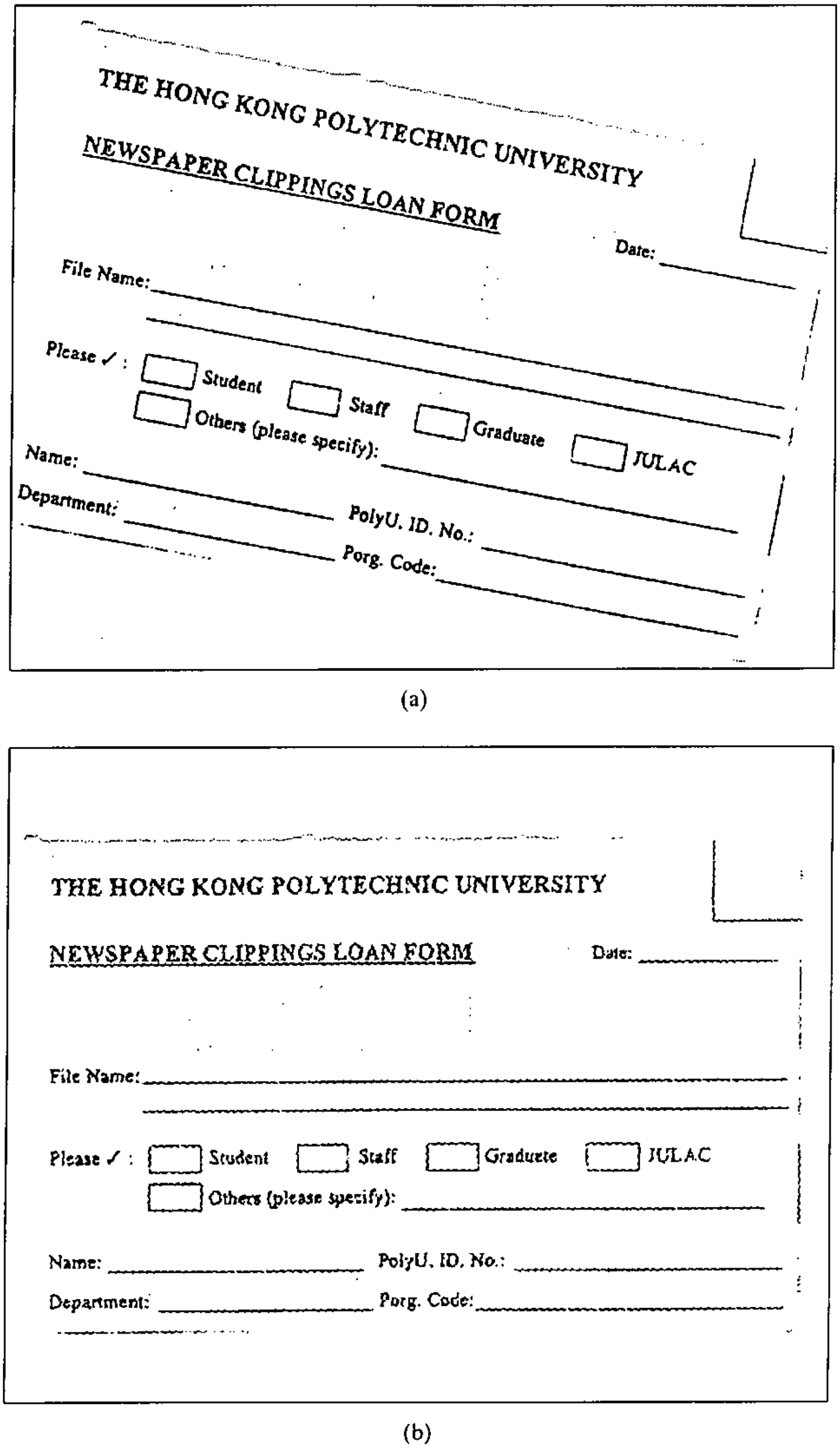
Because B(x,y) is binary, the multiplication in equation (2.9) can be implemented as logical AND. Thus the accumulated cross-correction function can be obtained without any multiplication. If the image is large, one may just need an area instead of the entire image for determining the skew angle.

Experimental Results

For equation 2.7, the image is rotated around the top left corner. In order to simulate rotation around the center of the image and preserve the same size, we have to transform the origin to the center of the image. Thus, equation 2.7 can be modified to:

$$\begin{aligned} X_{new} - X_C &= (X - X_C)\cos\theta - (Y - Y_C)\sin\theta \\ Y_{new} - Y_C &= (Y - Y_C)\cos\theta + (X - X_C)\sin\theta \end{aligned} \tag{2.11}$$

The result of skew detection and correction is shown in Fig. 2.6.



THE HONG KONG POLYTECHNIC UNIVERSITY

NEWSPAPER CLIPPINGS LOAN FORM

Date: ____

File Name: ____

Please ✓ : □ Student □ Staff □ Graduate □ JULAC

□ Others (please specify): ____

Name: ____ PolyU. ID. No.: ____

Department: ____ Porg. Code: ____

(a)

THE HONG KONG POLYTECHNIC UNIVERSITY

NEWSPAPER CLIPPINGS LOAN FORM

Date: ____

File Name: ____

Please ✓ : □ Student □ Staff □ Graduete □ JULAC

□ Others (please specify): ____

Name: ____ PolyU. ID. No.: ____

Department: ____ Porg. Code: ____

(b)

Fig. 2.6 Skew estimation and correction by line extraction method: (a) skewed image with an actual skewed angle of about 10°; (b) skew-corrected image with skewed angle estimated to be 10.8°.

2.6 Concluding Remarks

Several document image preprocessing tasks have to be performed for automatic form reading. This chapter discusses binarization, page segmentation, text block and line extraction, and skew estimation and correction. In binarizing a document images, we used two thresholds that were determined from the histograms of a local contrast measure and pixel intensities. Page segmentation is carried out to the binarized image and segmented blocks are further classified into three classes, text, lines and others, using three if-then rules. Lines are then used to determine the skewed angle of the image and, if necessary, a skew correction is carried out. Character segmentation and recognition will be performed on the extracted text blocks, which will discussed in next chapter.

Chapter 3

Script Determination

3.1 Introduction

Text is the main component in most documents. It is not uncommon that a country or a region use two or more written language, for example, both Chinese and English are official languages in Hong Kong. It is very necessary to have an algorithm that can determine scripts so that the text can be inputted to an appropriate character recognition system.

Little research has been done on separating Chinese document images from English ones. Jain *et al.* [4] proposed a neural network solution to page segmentation and, as a byproduct, script determination. In their approach, a three-layer (two hidden layers) feed-forward neural network was trained to separate the text in Chinese from that in English. The approach achieved promising results when it was tested on a couple of document images with Chinese text arranged in columns. However, in their approach, arbitrarily choosing training and test blocks from the whole image is neither cost-effective nor focusing on the problem. The neural network trained using such training blocks may not reliable in dealing with document images of different appearances, in particular, if the text region is small such as short text lines in form images. Moreover, computation is intensive due to using a large neural network with two hidden layers.

In this thesis, a robust, artificial neural network based script determination approach is proposed which can cope with different fonts, sizes, styles and darkness of text in document images. The Artificial Neural Network (ANN) approaches mainly have the following advantages:

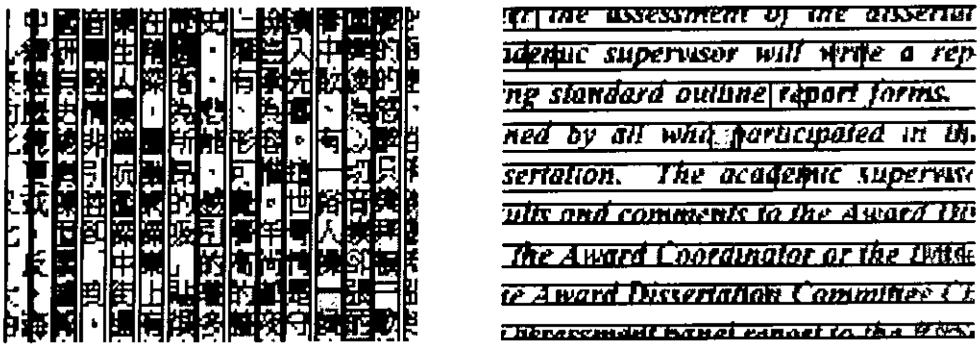
1. Massive parallelism: Basically, a neural network is composed of large amount of autonomous neurons interconnected by synapses. Each neuron can simultaneously perform its own functions. Thus, the ANNs are often implemented by massively parallel architectures.
2. High noise tolerance: Many proposed ANN modules have some degree of noise tolerance. In other words, these modules can process those data which are incomplete or noisy with graceful performance degradation.
3. Adaptability: ANNs can automatically adjust their internal parameters (e.g. synapse weights, neuron bias) to adapt them to various applications and also to improve their performance.

Texts of different scripts have different structures of fundamental elements (letters, characters, etc.) Chinese characters are usually square-like and more complex than English letters. This textual difference can be learned by training a neural network using the training samples chosen from text lines.

In this chapter, we will discuss preprocessing algorithms and how to derive the masks for feature extraction by training a three-layer feedforward neural network. Experimental results of using a smaller two-layer feedforward neural network for script determination are also reported in this chapter.

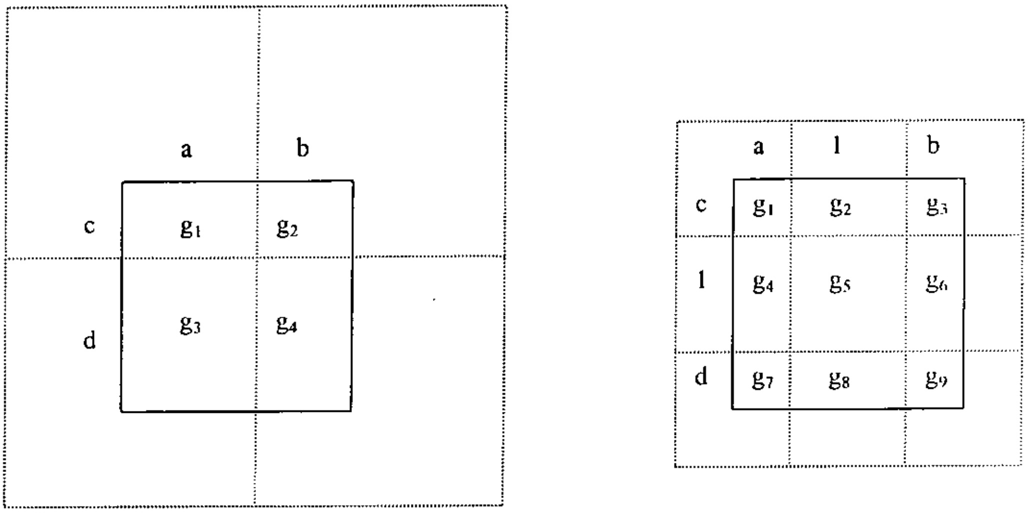
3.2 Neural Network Derived Feature Extractors

Before deriving feature extractors, several preprocessing steps are carried out to extract training and test samples from text lines. A square window with the sides equal to the height of a line is placed on the text line, sliding from the left to right, to extract sample blocks. Fig. 3.1 shows the examples of document images after sample block extraction.



(a) (b)

Fig. 3.1 Document images after sample block extraction: (a) a Chinese document image arranged in columns; (b) an English document image.



(a) (b)

Fig. 3.2 Block normalization: (a) s < 1; (b) s > 1.

To be fed to a neural network, each extracted block is normalized to a standard 7 × 7 window. Assume that the size of a block is $n_y \times n_x$. The scaling factors of X-axis and Y-axis are $s_x = n_x / 7$ and $s_y = n_y / 7$, respectively. In our system, we set $n_y = n_x = n$, hence $s_x = s_y = n / 7 = s$. In both expansion when $s < 1$ and shrinking when $s > 1$, we used the same scheme to normalize a block into a 7 × 7 window. Fig. 3.2 (a) and (b) depict the block normalization mapping for $s < 1$ and $s > 1$, respectively, where dotted lines represent the original grids and solid ones the grids in the 7 × 7 window. When $s < 1$ as shown in Fig. 3.2 (a), the average pixel value of the solid block corresponding to a pixel in the normalized 7 × 7 window is given by

$$g_{av} = \frac{acg_1 + bcg_2 + adg_3 + bdg_4}{(a+b)(c+d)} \tag{3.1}$$

where $(a+b) = (c+d) = s$ and g_i $(i=1,2,3,4)$ are gray scales of the four neighboring pixels in the original $n \times n$ image block. When $s > 1$, we may get a partition map as shown in Fig. 3.2(b). The average pixel value is then given by

$$g_{av} = \frac{acg_1 + cg_2 + bcg_3 + ag_4 + g_5 + bg_6 + adg_7 + dg_8 + bdg_9}{(a+1+b)(c+1+d)} \tag{3.2}$$

where $(a+1+b)=(c+1+d)=s$ and g_i $(i=1,2,\ldots,9)$ are gray scales of the nine neighboring pixels in the original $n \times n$ image block. Note that this algorithm is equally applicable to normalize rectangular blocks.

Images may have different gray scale ranges. Therefore, we have to normalize the input values so that the foreground and background pixels of images with different gray scales will lie in the same range. In our system, the obtained gray scale pixel intensities in the standard window are normalized bilinearly before they are fed into a neural network. They are normalized linearly from 0.0 to 0.5 if they are smaller than the binarization threshold (*T*) which is determined by histogram or from 0.5 to 1.0 if they are greater than the threshold. As shown in Fig. 3.3, the darkest pixel(s) have a normalized value of 0.0, the brightest pixel(s) a value of 1.0, and the pixels with the gray scale equal to the threshold have a value of 0.5.

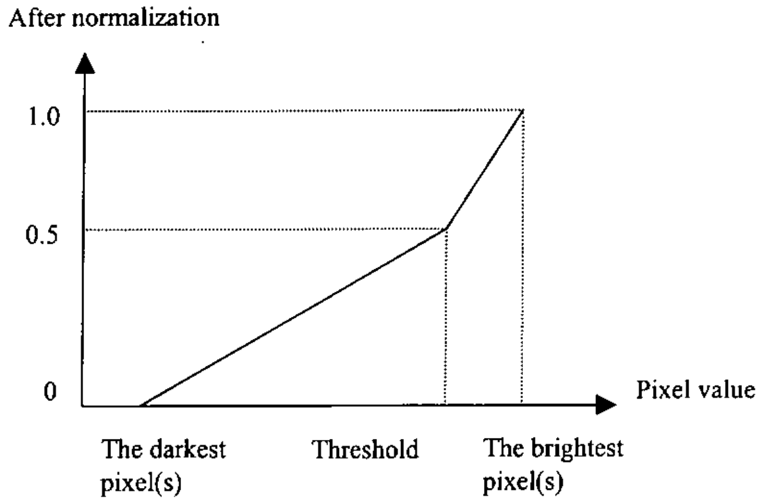


Fig. 3.3 Bilinear normalization of gray scale pixel values

As discussed in [4], a three-layer (two hidden layers) feedforward neural network is sufficient for image classification and it is employed to derive a set of masks for extracting features to perform script determination. The normalized 7 × 7 training blocks are used to train the neural network. To improve the performance, each of the training blocks is labeled as one of the three classes, Chinese, English, and blank where blank is to label those blocks of smaller than seven darker pixels based on the same binarization threshold *T*. A schematic diagram of the neural network is shown in Fig. 3.4. The input layer of the neural network contains 49 nodes receiving the inputs from the normalized pixels values in the standard 7 × 7 window. Fifteen nodes are used in the first hidden layer. The number of nodes in second layer is determined by try-and-error and it is found that 20 hidden nodes gave the highest accuracy. There are three nodes in the output layer that correspond to the three classes.

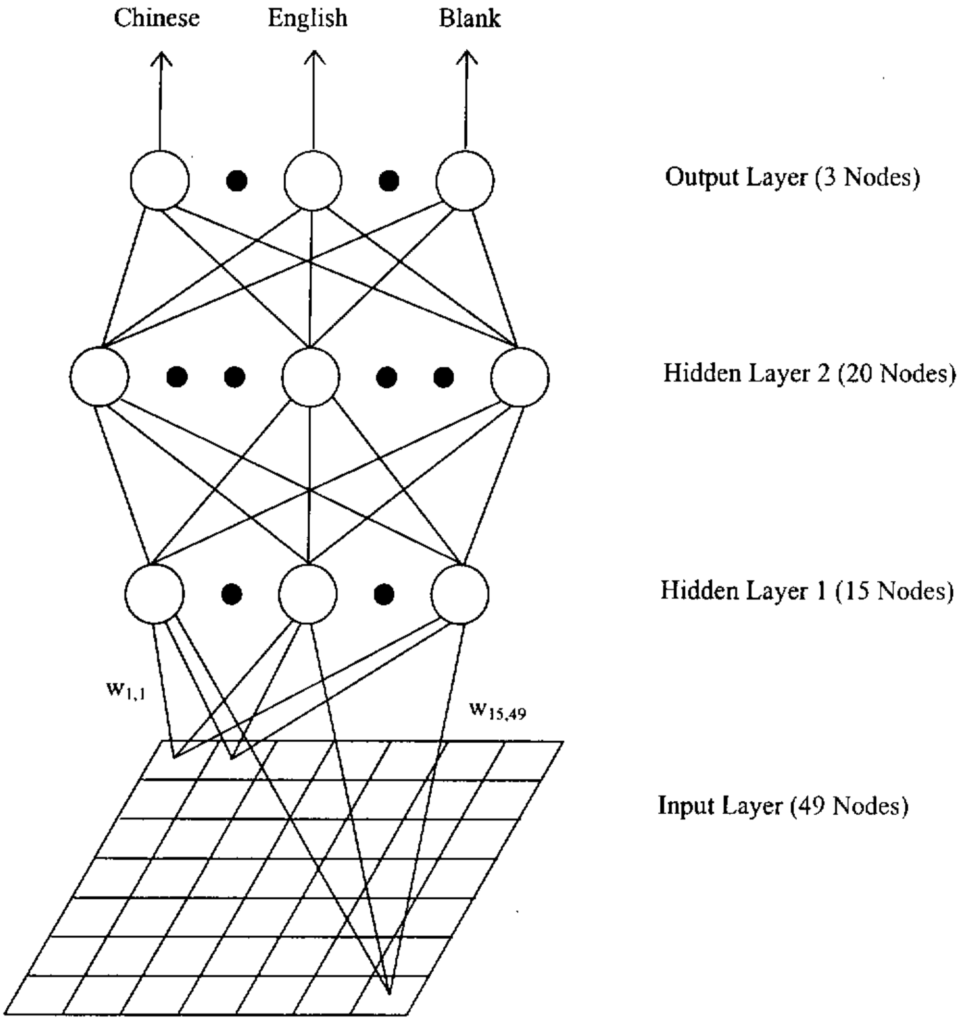


Fig. 3.4 Schematic diagram of the neural network for deriving the feature extractors

The neural network is trained with the back-propagation algorithm. During the training, the connection weights of the network are adjusted to minimize the classification error. A set of optimal weights are finally formed after a number of iterations of the training. Let the weight vector for node i in the first hidden layer be W_i. We have

$$W_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,49})^T \tag{3.3}$$

where $w_{i,j}$ is the connection weight from the jth pixel in the window to the ith node in the first hidden layer. We can interpret each weight vector between the input pixels and a node in the first hidden layer as a mask. In total, there are 15 masks corresponding to the 15 nodes in the first hidden layer. Let the pixel value of the jth pixel be g_j. The output of the ith node of the first hidden layer, o_i^1, is given by

$$o_i^1 = f(\sum_{j=1}^{49} w_{ij} g_j) \tag{3.4}$$

where $f()$ is the sigmoid function. To compute the output of each hidden node, 49 multiplication of two floating-point numbers plus $f()$ are required, which is a big computational burden. We found that almost all the weights were within [-2, 2]. In order to reduce the computational complexity, each of the coefficients of the 15 masks (weights) is approximated to the nearest one in a set of 17 pre-defined values as given below:

(-2.0, -1.75, -1.5, -1.25, -1.0, -0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0)

It is easy to verify that each number in the above list is the folds of 2^{-2}. Even in the worst case, only two shift operations and two additions are required to compute $w_{ij}g_j$. For example, if the approximated weight $w_{ij} = -1.75$, then

$$w_{ij} g_j = -1.75 g_j = -(g_j + 2^{-1} g_j + 2^{-2} g_j) \tag{3.5}$$

-0.25	-1.00	-0.50	0.00	-0.25	1.25	0.75
-0.50	-1.00	0.00	0.25	0.75	0.25	0.00
0.25	-0.25	-0.25	0.00	0.25	0.50	0.50
0.50	-0.50	0.00	-0.75	0.00	0.25	0.25
-1.00	0.00	-0.25	0.00	0.25	-0.50	-0.75
0.00	-0.75	0.75	0.00	0.75	-0.50	1.75
0.25	-0.50	-1.75	-1.25	0.25	-0.25	1.25

0.75	0.50	0.75	0.00	-0.50	-0.25	1.00
-0.25	0.00	-0.75	-0.75	0.00	0.00	-0.50
0.25	0.50	0.00	0.25	1.50	0.50	0.25
-0.75	-0.50	-1.00	-0.75	0.50	-0.25	-0.75
0.75	1.00	1.00	-0.25	0.75	0.75	0.00
-0.75	-0.25	-0.25	-0.75	-0.25	-1.75	-2.00
0.50	0.25	0.75	-0.25	-0.25	-0.25	-1.75

1.00	-0.25	0.50	-0.25	0.00	-0.25	0.75
-0.50	-1.00	-0.75	-0.50	-0.75	-0.25	-0.25
0.00	0.00	-0.25	-0.25	0.00	0.25	0.25
-0.50	0.00	-1.00	-0.25	-1.00	-0.25	-0.25
-0.75	-0.75	-0.75	0.00	-0.50	0.25	0.00
0.00	0.25	1.50	0.00	-0.50	-0.25	0.50
-0.25	0.00	1.00	-0.75	-0.75	0.25	0.50

0.25	1.00	0.50	0.25	1.00	0.50	-0.75
2.00	1.50	1.50	1.00	1.75	1.00	0.75
-1.75	-1.50	-0.25	0.75	0.50	-0.25	-1.50
-0.75	0.25	0.00	0.50	-0.25	-0.25	-1.00
0.75	1.00	0.00	-0.50	-0.75	1.00	-0.75
-2.00	-0.75	-0.75	-0.50	-0.25	0.25	-2.00
-1.75	0.25	1.00	0.50	0.50	0.00	-1.75

0.25	0.75	-0.25	0.00	0.00	-0.75	-0.25
-0.50	0.00	0.25	-0.75	0.50	0.00	0.00
-1.50	0.00	0.25	-1.50	0.50	0.75	1.25
-2.00	-0.25	0.25	-1.25	0.00	-0.25	1.25
-0.75	-0.25	0.25	-0.50	0.25	0.00	1.00
-0.50	0.00	0.00	-0.25	0.00	0.25	-0.25
0.00	-0.75	0.00	0.75	0.00	0.25	-1.25

0.50	-0.50	-0.25	-0.75	0.50	2.00	0.75
0.25	-0.25	-0.25	-0.50	-0.75	0.25	-0.75
0.25	0.50	-0.25	0.25	-0.50	1.25	-0.75
-1.00	-0.50	-1.00	-1.00	-1.00	0.00	-1.00
-0.50	0.50	0.75	-0.75	0.50	1.50	-0.75
-0.75	-0.25	0.75	-0.25	-0.50	2.00	1.25
-0.50	-0.50	0.25	-0.50	0.00	0.50	1.00

-1.50	0.50	0.00	-0.50	0.25	0.50	-1.00
-0.50	0.75	0.50	0.75	0.25	0.75	-0.75
-0.75	0.00	0.50	0.50	0.25	0.00	-1.75
-0.50	0.50	0.50	0.25	0.25	0.75	0.25
0.50	0.25	0.25	0.25	0.25	0.00	-0.25
-0.50	0.25	-1.00	0.75	-0.25	0.75	0.25
-0.50	1.25	0.25	0.75	0.50	1.50	0.00

1.00	1.00	0.25	0.00	-0.25	-0.50	-0.25
0.00	0.50	0.25	-0.50	-0.50	-1.00	-0.50
-0.25	0.00	0.00	-0.50	0.00	-0.75	-0.25
0.00	0.00	0.25	-1.50	-0.50	-1.00	-0.25
0.50	1.50	1.50	-0.25	0.25	-0.25	-0.25
-0.75	1.00	1.00	-0.50	-1.00	-0.75	-0.50
0.50	0.75	0.75	0.25	-0.25	-0.25	-1.25

1.25	0.50	0.50	-0.50	0.00	-0.75	0.25
0.75	0.25	1.25	-0.25	0.25	0.50	-0.50
-0.50	-0.25	1.75	-1.00	0.25	0.25	-1.00
-1.00	-0.50	2.00	-0.25	1.00	0.50	-1.25
-0.75	-1.00	0.75	-0.25	0.75	0.50	-0.50
0.25	0.00	0.75	0.25	0.25	0.50	0.50
-0.25	-0.75	0.00	0.50	0.25	0.25	0.00

-1.25	-1.00	-0.75	0.50	0.75	0.75	-1.50
-1.00	0.50	0.00	1.00	0.50	-0.25	-1.50
-0.25	-0.50	0.00	1.00	0.50	-0.50	-1.25
-0.75	-0.50	0.25	1.00	1.25	0.25	-0.50
0.50	0.50	0.00	0.75	0.50	0.75	0.25
-1.75	-2.00	-1.25	-0.50	0.75	0.25	-1.00
-1.25	-0.50	0.50	1.25	2.00	1.25	0.25

-0.75	-0.25	0.50	0.75	0.25	0.25	0.25
0.25	0.50	-0.50	-0.50	0.25	0.00	-0.50
1.25	1.00	1.25	-0.25	0.75	0.75	1.00
-1.50	-0.75	-0.50	-1.25	-0.50	-0.50	-1.25
1.50	1.25	2.00	0.25	1.50	2.00	1.50
-2.00	-0.25	-1.00	-1.50	-1.25	0.50	-0.25
-0.50	-1.25	-0.50	-1.00	-1.00	0.75	-0.25

0.25	0.75	-1.75	-2.00	-1.00	0.25	0.00
0.00	0.25	-0.50	-0.50	0.00	0.50	0.25
0.25	0.25	-0.25	-1.25	0.50	1.00	1.50
0.25	1.00	0.50	-0.25	0.25	0.25	0.50
-1.00	0.00	-0.50	-0.50	0.00	1.25	-0.75
0.50	0.75	-1.25	-0.75	-0.25	0.50	-0.75
0.75	1.25	-0.75	-0.25	-1.25	1.00	0.50

0.25	-0.75	-0.25	0.25	-0.75	-0.75	-0.25
0.50	0.75	0.75	1.25	-0.25	-0.25	-0.25
0.00	0.75	0.00	0.25	-0.50	-0.75	-0.50
0.50	0.00	0.00	0.25	0.00	0.00	0.75
0.75	0.25	-0.25	0.25	-1.50	-0.50	0.75
-0.50	0.25	0.50	0.50	-1.25	-0.50	-0.25
-0.25	0.75	0.50	0.50	0.00	-0.75	-0.50

0.75	0.25	0.00	0.25	0.25	0.00	0.75
1.00	1.00	1.25	0.75	0.50	1.00	1.25
0.00	-0.25	0.75	0.75	-1.00	0.75	0.50
0.00	-1.75	0.25	0.75	-0.75	-1.75	-0.25
0.25	-1.00	-0.75	0.50	0.50	0.75	0.50
-1.25	-1.00	-1.25	-1.75	-1.50	-0.50	-0.50
-0.25	-0.25	-0.75	-2.00	-0.75	-0.75	0.50

0.50	1.50	1.00	0.25	1.00	0.75	-0.50
-1.50	-0.75	-0.50	-1.00	-2.00	0.00	0.00
-1.25	-0.50	-0.50	0.50	-0.75	-0.25	-0.50
-1.25	-1.50	-1.25	0.00	-1.25	-1.00	-0.50
0.75	0.75	0.00	0.25	1.00	0.00	0.50
0.75	0.00	-0.25	0.25	0.25	0.25	0.50
0.25	1.00	1.00	1.00	0.75	0.75	-1.00

Fig. 3.5 The fifteen masks derived from the trained neural network

As we know, it takes a considerable amount of time to perform the multiplication of two floating-point numbers. Therefore the substantial saving in the computation can be achieved if we approximate the weights in this way. Fifteen approximated masks are shown in Fig. 3.5. Due to the approximation introduced in deriving the masks and the omitting of $f()$, a smaller two-layer feedforward neural network with the extracted features as the input is trained to perform the script determination.

3.3 Script Determination Using a Smaller Neural Network

To perform script determination, a two-layer (one hidden layer) feedforward neural network is utilized. By using the 15 masks we obtained, 15 features can be extracted. These features are used as the input of the neural network. The optimal number of nodes in the hidden layer can be determined by preliminary experiments in order to improve the classification performance while reducing the computational complexity. After testing, we use 10 nodes in the hidden layer, a half of those used in the second hidden layer of the first neural network employed for deriving masks. Again three output nodes are used for three classes. In the first neural network, we have $15 \times 20 + 20 \times 3 = 360$ weights from the first hidden layer up. However, in the retrained smaller neural network, we have $15 \times 10 + 10 \times 3 = 180$ weights which accounts for further 50% saving in the computation.

For each texture image or text line, a number of sample blocks are randomly extracted and fed to this neural network classifier. The script of the text image is then determined by a simple majority vote among sample blocks.

3.4 Experimental Results

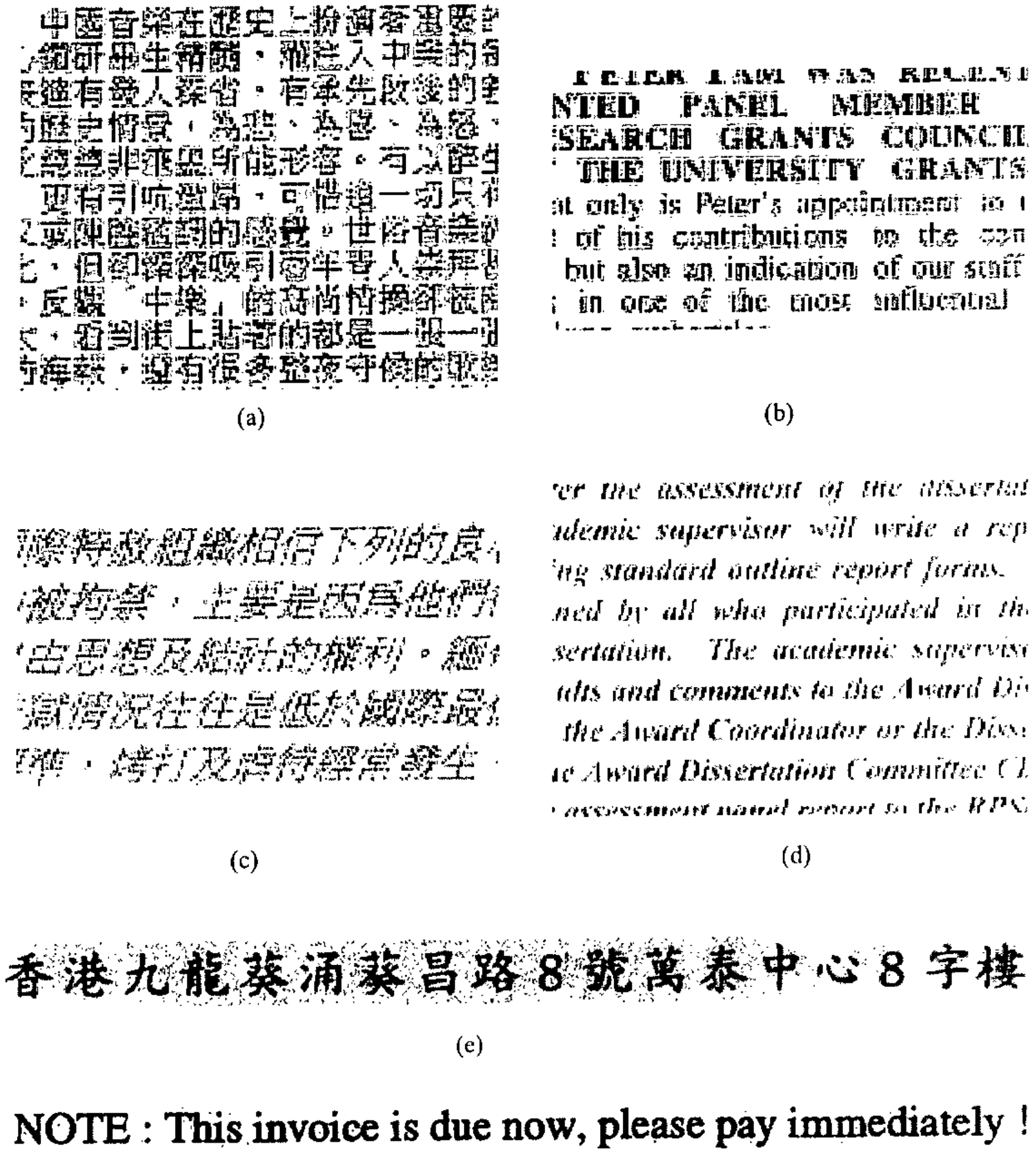
The script determination algorithm has been applied to 40 document images (a half of them are English document images and the rest are Chinese ones) and 90 text lines from several form images of different appearances. These documents were scanned with a resolution of 100 *dpi* using a Hewlett Packard ScanJet IIcx desktop scanner.

Ten arbitrarily selected English document images, ten Chinese document images, and 43 text line images were used as a training set. The rest of document images and text lines were used as a test set. We extracted totally 8600 sample blocks for the training. These blocks were fed into a 49-15-20-3 neural network to derive a set of 15 masks. To reduce the computational complexity, the coefficients of the 15 trained masks were approximated to the nearest 17 pre-defined values within [-2, 2] as discussed in Section 3.2. By using the 15 masks, 15 features were extracted and used to train a smaller 15-10-3 neural network for the script determination.

To evaluate the classification performance on the independent test images, we also extracted 200 sample blocks from each test image to test the trained neural network. The test results for both the training images and test images are shown in Table 3.1.

Table 3.1: Test results on the training and test document images and text lines from several form images (NI: the number of images; CCI: the number of correctly classified images).

	Training Set		Test Set	
	NI	CCI (%)	NI	CCI (%)
Chinese Documents	10	10 (100)	10	10 (100)
English Documents	10	10 (100)	10	10 (100)
Chinese text lines	21	21 (100)	23	21 (91.3)
English text lines	22	21 (95.5)	24	24 (100)
Overall	63	62 (98.4)	67	65 (97.0)



中國音樂在歷史上扮演著重要
鑽研學生精髓，融注入中樂的
延續有幾人探皆，首承先啟後的
與歷史背景，為悲、為喜、為怒、
之感覺非筆墨所能形容。可以說是
更有引吭盈場，可惜這一切只有
之要陳腔濫調的感覺，世俗音樂
亡，但卻深深吸引當年青人共鳴
，反觀「中樂」能高尚情操卻被
之，看到街上貼著的都是一張一張
的海報，還有很多整夜守候的歌迷

(a)

PETER LAM WAS RECENT
NTED PANEL MEMBER
SEARCH GRANTS COUNCIL
THE UNIVERSITY GRANTS
at only is Peter's appointment to
of his contributions to the con
but also an indication of our staff
in one of the most influential

(b)

際特赦組織相信下列的良
被拘禁，主要是因為他們
由思想及結社的權利。經
罰情況往往是低於國際最
準，虐打及虐待經常發生

(c)

er the assessment of the dissertat
ademic supervisor will write a rep
ing standard outline report forms.
ned by all who participated in th
sertation. The academic supervis
ults and comments to the Award Di
the Award Coordinator or the Diss
e Award Dissertation Committee (A
assessment panel report to the RPC

(d)

香港九龍葵涌葵昌路8號萬泰中心8字樓

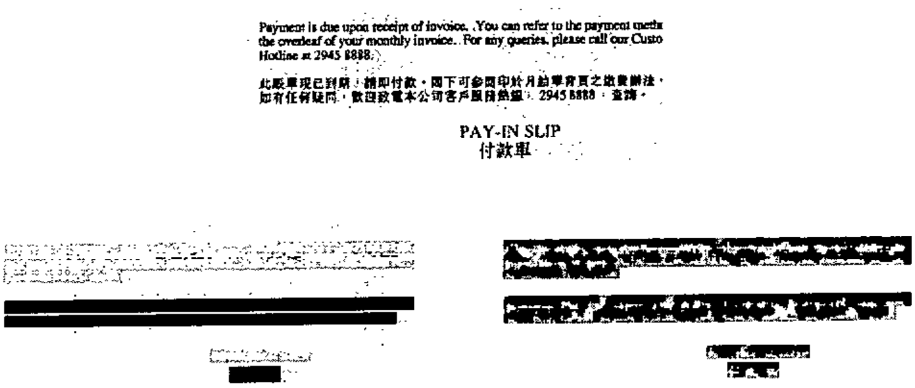
(e)

NOTE : This invoice is due now, please pay immediately !

(f)

Fig. 3.6 (a) A Chinese document image used for training; (b) an English document image used for training; (c) a Chinese document image used for testing; (d) an English document image used for testing; (e) a text line image used for testing; (f) a text line image used for testing.

From the test results, we can see that more than 97% of the images in the training and test sets are correctly classified. It shows that this approach is quite reliable for the separation of the Chinese and English text. Fig. 3.6 shows some examples of images included in the experiments. Note that some images have significantly different appearances even in the same document. For example, the image shown in Fig. 3.6(b) has two different fonts (bold and plain) of characters with different sizes where one nearly doubles the size of the other. In addiction, we have also compared the results by applying our algorithm and the algorithms proposed by Jain and Yu on some form images. Fig. 3.7 shows the results of both algorithms.



Payment is due upon receipt of invoice. .You can refer to the payment meth
the overleaf of your monthly invoice.. For any queries, please call our Custo
Hotline at 2945 8888.

此賬單現已到期，請即付款。閣下可參閱印於月結單背頁之繳費辦法。
如有任何疑問，歡迎致電本公司客戶服務熱線：2945 8888，查詢。

PAY-IN SLIP
付款單

Fig. 3.7 (a) A sample form image; (b) Result of our approach (black line: Chinese; gray line: English); (c) Result of Jain and Yu's method (black region: Chinese; gray region: English)

3.5 Concluding Remarks

In this chapter, script determination is performed to classify a text region in the document images into Chinese or English. A neural network approach has been applied for script determination. In this approach, two neural networks are employed. The first neural network is trained to derive a set of 15 masks that are used for extracting 15 features. The coefficients of masks are then approximated to the nearest pre-defined values within [-2, 2] for reduced computational complexity. The second neural network of a smaller size is trained with 15 extracted features as the input to perform the script determination of each sample block. The script of a texture image or a text line is determined by a simple majority vote among sample blocks. Experimental results show that 85 out 87 test images can be correctly classified. By using this algorithm, document images of two different languages, Chinese and English, can be identified. It is very useful and helpful for an automated document reading system where characters of different scripts need to be fed into different character recognition sub-systems.

Chapter 4

Fast Keywords Matching with DRNN

4.1 Introduction

We have to do lot of keywords matching when we use computers to edit documents, look up information, debug programs, glance over Internet, and much more. There are many mature techniques dealing with such matching in electronic documents. However, it is a very challenging task to fast matching the keywords in optical documents which are obtained by scanning printed documents. As it is the major channel to use printed documents to transfer information in the real life, it is much demanded to have a fast algorithm for direct keywords matching in optical documents. Fast keywords matching is also important for retrieval of document images.

It is not a difficult task to matching keywords in standard documents with the preset character font, size, spacing, and background. However, very often, we are exposed to a variety of documents that are very much different in appearances and styles. In the case of name cards, we have such a diverse collection in which name cards of different designs were printed in different fonts, sizes, background, and even in different scripts. It is a challenging task to perform character recognition on these documents partly because we cannot successfully segment characters. Very often, we have to deal with many connected and broken characters. Direct word recognition may be a short-cut solution for keywords matching.

Many recognition techniques have been developed by either using completion on words [52] or direct keywords recognition [53, 54]. Kuo *et al.* [53] have proposed the use of Hidden Markov Models for keywords recognition. In their approaches, 1D HMM is used for keyword recognition. Then it is extended to a pseudo 2D model. The 1D HMM approach achieves only 70% accuracy rate while the pseudo 2D HMM approach can achieve an accuracy of 99% when the words are in the same size, or 96% if the words are in different sizes.

In this chapter, we focus on a fast, robust and computationally less complex algorithm for keywords matching in various environments, such as different character fonts, spaces, sizes, languages, background, etc. In our approach, a Dynamic Recognition Neural Network (DRNN), a robust and flexible model with low computational complexity, is trained using a training set of pre-classified samples to perform this function. The block diagram of our approach is shown in Fig. 4.1.

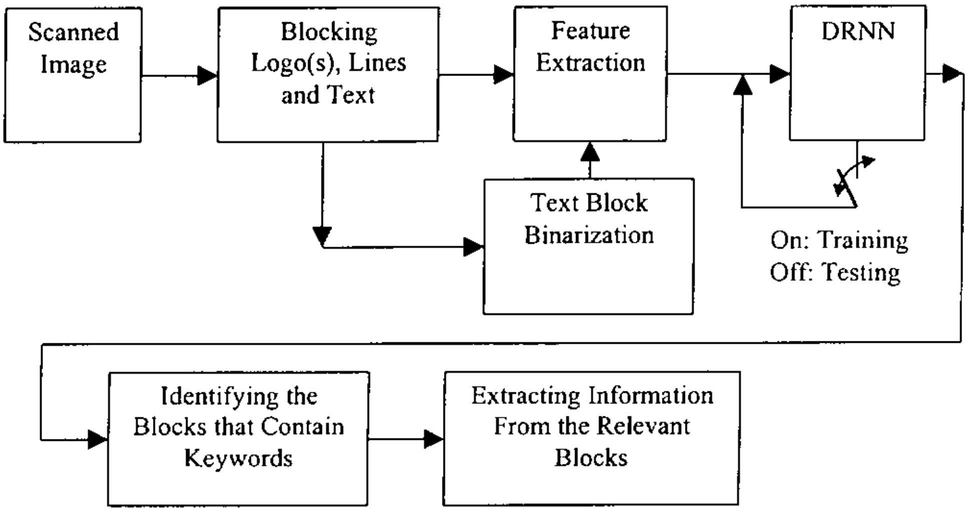

Scanned Image
Blocking Logo(s), Lines and Text
Feature Extraction
DRNN
Text Block Binarization
On: Training
Off: Testing
Identifying the Blocks that Contain Keywords
Extracting Information From the Relevant Blocks

Fig. 4.1 Block diagram of fast keywords matching

4.2 Feature Extraction for Keywords Matching

Preprocessing of a scanned document image for fast keywords matching consists of two steps: (1) blocking logo(s), text and lines and (2) feature extraction.

Projection profiles (x and y) and the run-length smoothing algorithm are used to block the text, logo(s) and lines as both algorithms have low computational complexity and are therefore suitable for fast processing. On the other hand, the DRNN model is flexible and robust so that the requirement for locating text blocks is not very strict in terms of precision.

The projection profile is used for determining the rough layout of a name card, that is, finding out whether it is arranged vertically or horizontally. Also the threshold of run-length smoothing is obtained from the projection profiles of an image. Let $im(x, y)$ be the intensity of a document image at location (x, y) and R denotes the area of the document image. The projection profiles can be obtained by

$$p(x) = \int_R im(x,y)dy \qquad (4.1)$$

$$p(y) = \int_R im(x,y)dx \qquad (4.2)$$

Where $p(x)$ and $p(y)$ are vertical and horizontal projection functions, respectively.

The run-length-smoothing algorithm is used to link all characters in a word together and to form a text block. For a 0's string, if the length of the string is smaller than a preset threshold (c), then the string will be replaced with 1's string.

Simple and effective features should be used as the fast realization is our main concern. Different sets of features are used to deal with different scripts. For English, projection profiles $p(x)$ and $p(y)$ are used because it is easy to extract and contains enough information for keywords matching. Since Chinese characters contain more strokes than English letters and we have a very large set of Chinese characters. It is difficult, if not impossible, to recognize Chinese characters based on projection profiles, especially when the character size is small. For Chinese, contour features are utilized. Let *bim*(*x*,*y*) be a binarized test block image pixel and R_t denotes the text block, *p*(*x*) and *p*(*y*) can be obtained by Equations (4.1) and (4.2) with *R* and *im*(*x*,*y*) replaced by R_t and *bim*(*x*,*y*) respectively. The contour features are defined as

$$ctb(x) = \sum_{y=1}^{n} g(bim(x,y)) \tag{4.3}$$

$$cbt(x) = \sum_{y=n}^{1} g(bim(x,y)) \tag{4.4}$$

$$clr(x) = \sum_{x=1}^{m} g(bim(x,y)) \tag{4.5}$$

$$crl(x) = \sum_{y=m}^{1} g(bim(x,y)) \tag{4.6}$$

Where *m* and *n* are the width and height in pixels of the text block respectively, and *ctb*(*x*), *cbt*(*x*), *clr*(*y*), *crl*(*y*) are top-bottom, bottom-top, left-right and right-left contour features respectively.

The function *g*(*bim*(*x*,*y*)) as defined as

$$g(bim(x,y)) = \begin{cases} 1 & im(x,y) \in S \\ 0 & im(x,y) \in T \end{cases} \tag{4.7}$$

Where *S* is a set of white pixels from one of image frames to the first black pixels it touches when the image is scanned on one of four directions, top-down, bottom-up, left-right, and right-left. *T* is the complement set of *S*, that is, $S \cup T = R$, *and* $S \cap T = \varnothing$. Figure 4.2 shows *S* and *T* sets for four scanning directions with black regions marking *S* sets.

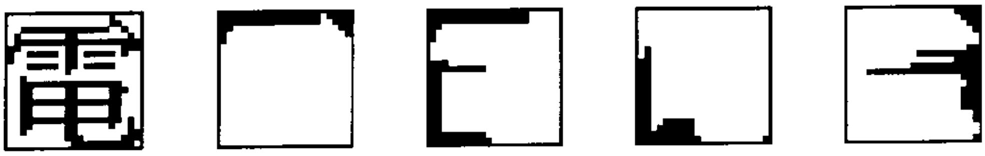


Fig. 4.2 Contour features of a Chinese character: (a) a Chinese character; (b) top-bottom contour feature; (c) left-right contour feature; (d) bottom-top contour feature; (e) right-left contour feature.

4.3 DRNN for Keywords Matching

Zhang *et al*'s study showed that the DRNN model had high flexibility and low computational complexity in dealing with the problem of time axis alignment in speech recognition [55]. For keywords matching, the font, size, spacing, style and many other conditions are variables, which introduces the similar problem as speech recognition. Therefore, we expect that the use of the DRNN model would improve the performance in keywords matching in document processing.

4.3.1 DRNN Architecture

In pattern recognition, a test pattern is classified to the closest prototype based on a distance measure,

$$d(X^{(i)},Y), \qquad i=1,2,\ldots,N \tag{4.8}$$

where vector $X^{(i)}=\left[x_1^{(i)},x_2^{(i)},\ldots,x_m^{(i)}\right]$ is the i-th prototype, vector $Y=\left[y_1,y_2,\ldots,y_m\right]$ denotes a test pattern to be recognized, and N is the number of prototypes. If a pattern is represented by a fixed-dimension vector, pattern matching can be performed straightforwardly. However, in some applications, the dimensionality of patterns is variable. Hence, we have

$$X^{(i)}=\left[x_1^{(i)},x_2^{(i)},\ldots,x_{p(i)}^{(i)}\right] \tag{4.9}$$

$$Y = \left[y_1, y_2, \ldots, y_q\right] \tag{4.10}$$

where $p(i)$ is the length of ith prototype and q is the length of a test pattern. In most cases, $p(i) \neq q$.

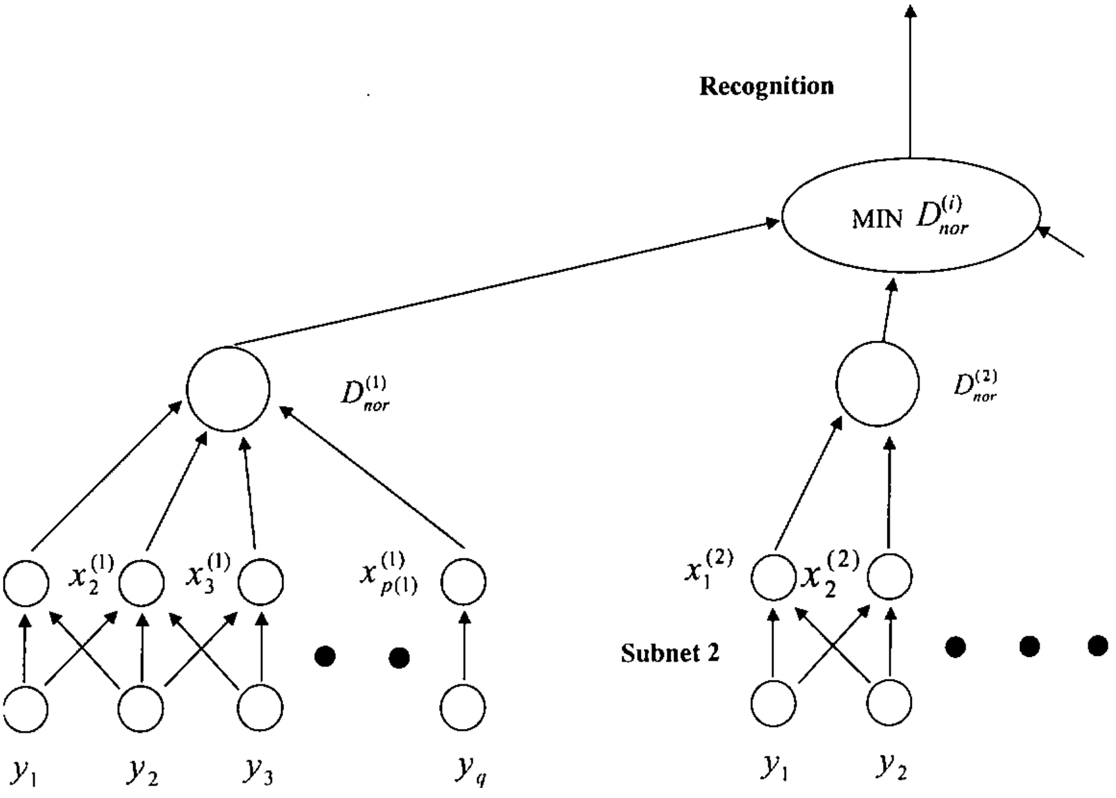


Fig. 4.3 The configuration of the DRNN

In our DRNN approach, Y is recognized as the i-th pattern if

$$i = \arg\min_{1<i<N} \left\{ d\left(X^{(i)} Y\right) \right. \tag{4.11}$$

Because $p(i) \neq q$, a nonlinear comparison is required between the two vector sequences of different lengths. The distance function designed should be incorporated with time alignment. This nonlinear relationship between the time series can be given by:

$$Z^{(i)} = T_W(Y) \tag{4.12}$$

where $Z(i) = z_1^{(i)}, z_2^{(i)}, ..., z_{p(i)}^{(i)}$ and T_W is a nonlinear reflecting array, which is based on the optimal time alignment between two vector sequences of different lengths. After a nonlinear conversion, the distance function is to compare two time series of the same length.

Consider Equations (4.8) and (4.12), we have:

$$d_W\left(X^{(i)}Z^{(i)}\right) = d_W\left(X^{(i)}, T_W(Y)\right) \tag{4.13}$$

1) The start point $x_1^{(i)}$ of prototype pattern $X^{(i)}$ is aligned to y_1 of test pattern Y. The algorithm has no strict aligning requirement for the end point. However, there is a fixed search radius that restricts the range of comparison. We choose the length of sequences to be compared as that of the shorter vector sequence of two, that is,

$$r = \min(p(i), q) \tag{4.14}$$

2) Compare every y_k, where k is from 1 to r, with $x_{k-s}^{(i)}, ..., x_k^{(i)}, ..., x_{k+s}^{(i)}$ to find the minimum distance:

$$D_{\min k}^{(i)} = \min_t\left\{\left|x_{(t)}^{(i)} - y_{(k)}\right|\right\} \tag{4.15}$$

where $t = k\text{-}s, \ldots\ldots, k, \ldots\ldots, k\text{+}s$ and s is a parameter to control the search range. | . | is an absolute operation.

3) Compute the normalized minimum distance (D_{nor}) between the test and each prototype pattern:

$$D_{nor}^{(i)} = \frac{1}{r}\sum_{k=1}^{r} D_{\min k}^{(i)} \tag{4.16}$$

4) Finally, the test pattern is classified to i-th prototype if

$$i = \arg\min_{1<i<N}\left\{d_W\left(X^{(i)}, T_W(Y)\right)\right\}$$

$$= \arg\min_{1<i<N} \{ D_{nor}^{(i)} \tag{4.17}$$

Where T_W is implicitly implemented by reflecting Y non-linearly.

4.3.2 DRNN Learning Rule

The DRNN learning is adaptive and accumulative. The detail of the learning algorithm is described as follows:

1) The first incoming input pattern sequence for each prototype is taken as the initial weights of the prototype. Two or more prototypes may be used for a class of patterns. The patterns that are very different from the other patterns in the same class are chosen as initial weights. Each class could include several sub-nets so that the neural network is more robust.

2) For any pattern from the training data, $Y = y_1, y_2, ..., y_q$ find the optimal matching trace $[D^{(i)}]$ with prototypes via Equation (4.15). The optimal matching trace is given by:

$$[D^{(i)}] = [D_{\min 1}^{(i)}, D_{\min 2}^{(i)}, ..., D_{\min k}^{(i)}, ..., D_{\min r}^{(i)}] \tag{4.18}$$

where r is the length of the shorter pattern of the two vectors.

3) According to the optimal matching trace $[D^{(i)}]$, the weights of the DRNN are adjusted by:

$$x_{\min k}^{(i)}(t+1) = \frac{\left[x_{\min k}^{(i)}(t) \times n_i + y_{\min k}\right]}{\left[n_i + 1\right]} \tag{4.19}$$

where n_i, is the number of modifications have been done to the ith prototype pattern.

4) If $p\ (i) < q$, we have

$$X(i)(t+1) = \left[x_1^{(i)}(t+1), x_2^{(i)}(t+1), ..., x_{p(i)}^{(i)}(t+1), y_{p(i)+1}, ..., y_q\right] \tag{4.20}$$

5) Go to step 2) until all of the training patterns have been presented to the neural network.

4.4 Experiments on Name Card Keywords Matching

The reason that we chose name cards to test our approach is that name cards contain rich information and have various styles in terms of arrangements, character fonts, line spacing, background with different patterns or colors, company logos, etc. Twenty-nine name cards of the people from different countries, such as China, Hong Kong, Canada, Australia, Singapore, and Japan, are used in our experiments. Examples of name cards we used are shown in Fig. 4.4.

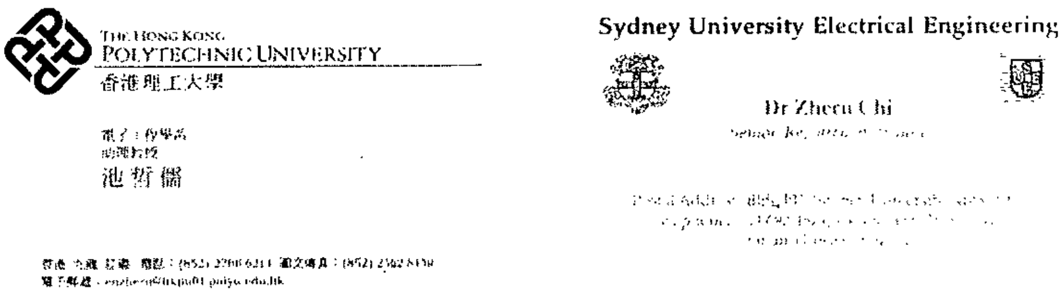


Fig. 4.4 Examples of name cards used in our experiments

In our experiments, the keywords Telephone and Facsimile were chosen as they are used frequently in daily life. Both of them have different appearance, e.g., Tel, TEL, Telephone, Fax, FAX, Facsimile. Fig. 4.5 shows some examples of keyword images. Therefore, the DRNN should include different prototypes for each keyword. Two or three training images are used for each prototype in the experiments. As discussed in Section 4.2.2, x and y projections were used for matching keywords in English and contour features were used for matching keywords in Chinese. Table 4.1 shows the experimental results on the training set of images of different keywords: Fax, Tel., Address, Pager, E-mail, Internet, etc. Table 4.2 shows the results on the test set of images that are different from those in the training set.

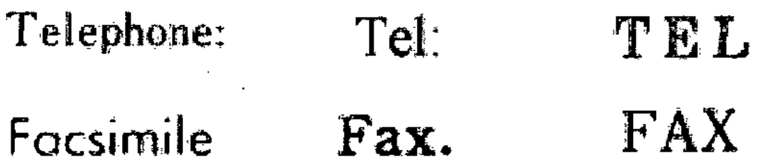


Fig. 4.5 Some examples of keyword images

Table 4.1. Keywords matching accuracy in the training set

Keywords	Tel	TEL	Telephone
Recognition Accuracy	100%		
Keywords	Fax	FAX	Facsimile
Recognition Accuracy	100%		

Table 4.2. Keywords matching accuracy in the test set

Keywords	Tel	TEL	Telephone
Recognition Accuracy	92%		
Keywords	Fax	FAX	Facsimile
Recognition Accuracy	91%		

We have also tested our approach on the keywords Telephone and Facsimile in Chinese words that include simple and complex characters. Table 4.3 shows the results on the training set of images of different Chinese keywords: 電話, 傳真, 地址, 傳呼, 電子郵件, etc. Table 4.4 shows the results on the test set of images.

Table 4.3 Chinese keywords matching accuracy in the training set

Keywords	電話 (Telephone)	电话 (Telephone in simple character)
Recognition Accuracy	100%	
Keywords	傳真 (Facsimile)	传真 (Facsimile in simple character)
Recognition Accuracy	100%	

Table 4.4 Chinese keywords matching accuracy in the test set

Keywords	電話 (Telephone)	电话 (Telephone in simple character)
Recognition Accuracy	90%	
Keywords	傳真 (Facsimile)	传真 (Facsimile in simple character)
Recognition Accuracy	89%	

We have also compared our approach with direct template match. The experimental results are shown in Table 4.5.

Table 4.5. Recognition accuracy of the DRNN approach compared with templete match

Keywords	Tel	TEL	Telephone
DRNN	92%		
Template Match	72%		
Keywords	Fax	FAX	Facsimile
DRNN	91%		
Template Match	70%		

4.5 Concluding Remarks

Keywords matching is very useful in optical document processing. A DRNN model for fast and robust keywords matching is presented in this thesis. The proposed approach can deal with ever-changing environment to capture keywords in OCR documents of different layouts, various character fonts and sizes, and different spacing. Moreover, the nature of accumulative learning of the DRNN model make it easily adapted to new environments. Our approach could achieve 90% correct matching rate when it was test on keywords matching of 29 name cards, which is much better then direct template match approach.

Chapter 5

Character Recognition

5.1 Introduction

Character recognition is a very active research area. It is one of the longest histories of commercial products for pattern recognition applications [7] and is a famous problem that Artificial Neural Networks (ANNs) have been applied to. Character recognition has found many useful applications in computerized processing of printed/handwritten. Many techniques and applications have been reported in the literature [8,31,43].

Character segmentation is a necessary step for character recognition although in some algorithms both the segmentation and recognition are tightly coupled. Lee *et al.* [8] proposed a methodology for gray-scale character segmentation and recognition which is composed of three steps: determination of character segmentation region, search for nonlinear character segmentation paths by multi-stage graph search algorithm, and confirmation of the nonlinear character segmentation paths and character recognition results. In his methodology, the candidate segmentation points could be found by using topographic features and projection profiles of gray-scale images. Nonlinear character segmentation paths and character recognition results could then be found by adopting a recognition-based segmentation scheme.

Character segmentation is a technique by which a text line or words is segmented into individual characters. It is a critical step because incorrectly segmented characters are not likely to be correctly recognized. The difficult cases in character segmentation are broken characters and touching characters. Moreover, the complexity of character segmentation stems from:

1. A wide variety of fonts and rapidly expanding text styles;
2. poor-quality images especially those generated from sparse dot matrix printers;
3. poor binary images resulted from the thresholding process in the binarization step.

All these can cause the problems of broken and touching characters. Broken and touching characters are responsible for the major errors in automatic reading of both machine-print and hand-written text.

Character recognition is a difficult problem due to a large number of variations in font sizes and styles. In the past research, many techniques of character recognition have been proposed [7,33,56,57]. One of the most popular approaches is the feature matching method. In general, a feature matching technique needs to extract a sufficient number of features from a character image in order to achieve correct recognition. If the extracted features were incomplete or noisy, then it would most likely end up with a wrong recognition. Besides, some dynamic programming or relaxation techniques are often used to achieve the optimal matching. However, these methods often slow down the recognition speed. Recently, the artificial neural networks have attracted more and more attentions in pattern recognition circle [43].

Many researchers have tried to use ANN as an alternative approach for character recognition and achieved satisfactory results [33,44,45].

In this thesis, we use an artificial neuron network to recognize English characters, digits and some symbols. Text characters in a binary image will first go though the process of segmentation. Then each segmented character will be normalized to a standard size image in which, in total, 292 combined intensity and skeleton-based features are extracted. These features will then be used for classification.

5.2 Character Segmentation

Most character segmentation algorithms were developed for binary text images. Character segmentation in binary images is often based on the fact that bit patterns of characters can be separated by scans containing no 'black' bits. However, this method is clearly not adequate to separate touching characters. In this project, an algorithm based on vertical projection and a peak-to-valley function was adopted for segmenting characters.

5.2.1 Algorithm Based on Vertical Projection Functions

The vertical projection $V(x)$, which is served as an important feature in segmentation, is widely used in character segmentation. The vertical projection is the histogram obtained by counting the number of black pixels in each vertical scan. If the characters are well separated, $V(x)$ should have zero values between characters and character segmentation can be accomplished directly from he vertical projection function on each text line. Fig. 5.1 shows a text line in which the characters can be separated straightly by vertical projection function.

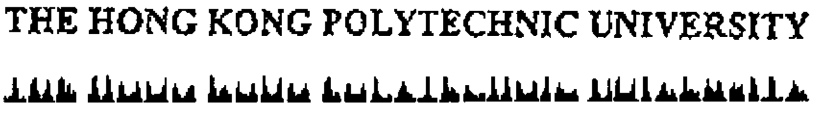


Fig. 5.1 Vertical projection function of a text line

Many text images do not have such a perfect condition and therefore can not be correctly segmented by using the vertical projection function only. Thus, we have to find another algorithm to deal with the touching or broken characters. In our approach, we deliberately used a smaller threshold for image binarization so that the binarized images are usually quite dark. As a result, we only need to deal with the touching characters but not broken characters.

Since most of the characters in documents or forms are in proportional fonts, the main issue involved in this problem is to find the break locations, which is a more challenging problem than that in fixed-pitch fonts. Pavlidis *et al* [58] used recognition algorithms to identify single-character from multiple-character blobs. They argued that a joined two characters must have a sharp minimum in the vertical projection, hence they proposed a function, the Peak-to-Valley function, for finding breaking points within merged characters. The function $V(x)$ to $V(x+1)$, namely, $PV(x)$:

$$PV(x) = \frac{V(x-1) - 2 \times V(x) + V(x+1)}{V(x)} \tag{5.1}$$

where $V(x)$ is the vertical projection function, x is the current position. The peak-to-valley function emphasizes local minima in the vertical projection and works quite well in finding breakpoint locations within touching characters.

Before splitting, we assume that the width of a character is between $H/4$ to H, where H is the height of a single-line multi-character block. That means, if the splitting region has a width of larger than H, the splitting process will be continued until the width is smaller than $H/4$. We first calculate the minimum value of the vertical projection, $V(x)_{\text{min}}$ and the maximum value of the peak-to-valley function, $PV(x)_{\text{max}}$. The algorithm of the splitting process is described as follows:

```
1.  For $a = PV(x)_{\text{max}}$ to 1 at a step of -1, do:
2.  for $x$ from the leftmost position to rightmost position, do:
3.  if [ $x$ is in the splitting region ] and [ $V(x) = V(x)_{\text{min}}$ ] and [ $PV(x) = a$ ],then
    split there and set $(x-H/4)$ to $(x+H/4)$ to non-splitting region
4.  end loop $x$
5.  end loop $a$
```

After this splitting process, there still exit some regions that have a width of larger than H. Therefore, we need to do the splitting second time according to the following steps:

```
1.  for $x$ from the leftmost position to rightmost position, do:
2.  if [ $x$ is in splitting region ] and [ $PV(x) > 2$ ] , then split there and set $(x-H/4)$
    to $(x+H/4)$ to non-splitting region
3.  end loop $x$
4.  if the split region contains less than 5 black pixels, neglect it
```

An example of the splitting process is shown in Fig. 5.2.

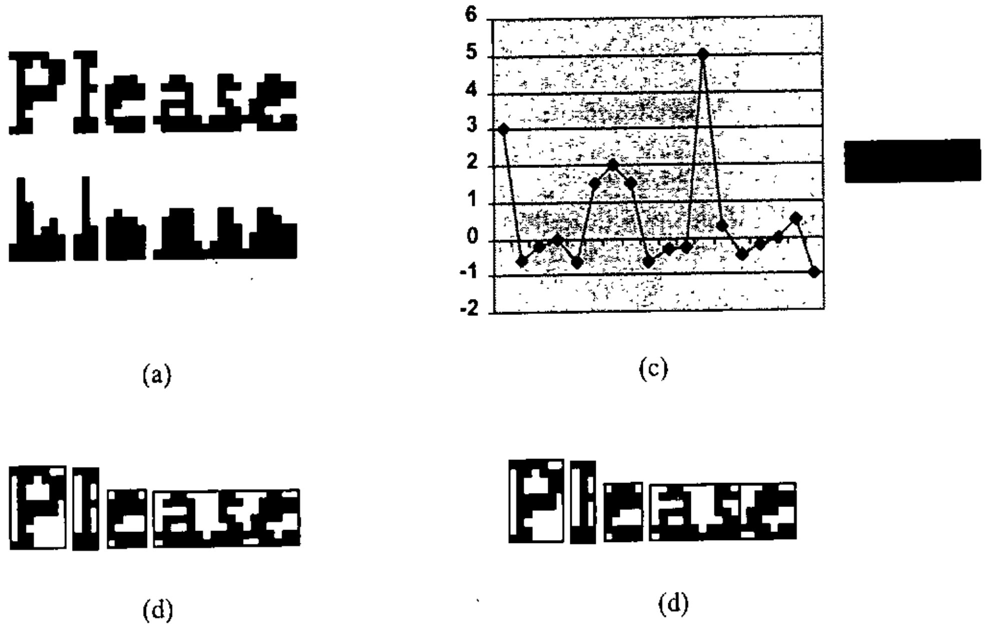


Fig. 5.2 Character segmentation: (a) a binarized image and the corresponding vertical projection function; (b) character segmentation based on the vertical projection function; (c) Peak-to Valley function in the connected region, i.e. the characters 'ase'; (d segmentation result using Peak-to-Valley function

5.2.2 Experimental Results of Character Segmentation

By using the above proposed algorithm, about 86% of the characters can be correctly segmented in form images with 100 *dpi* scanning resolution. It is also found that a much higher accuracy can be achieved if the images with higher scanning resolution of 150 *dpi* are used. The result of segmentation of a form image is shown in Fig. 5.3.

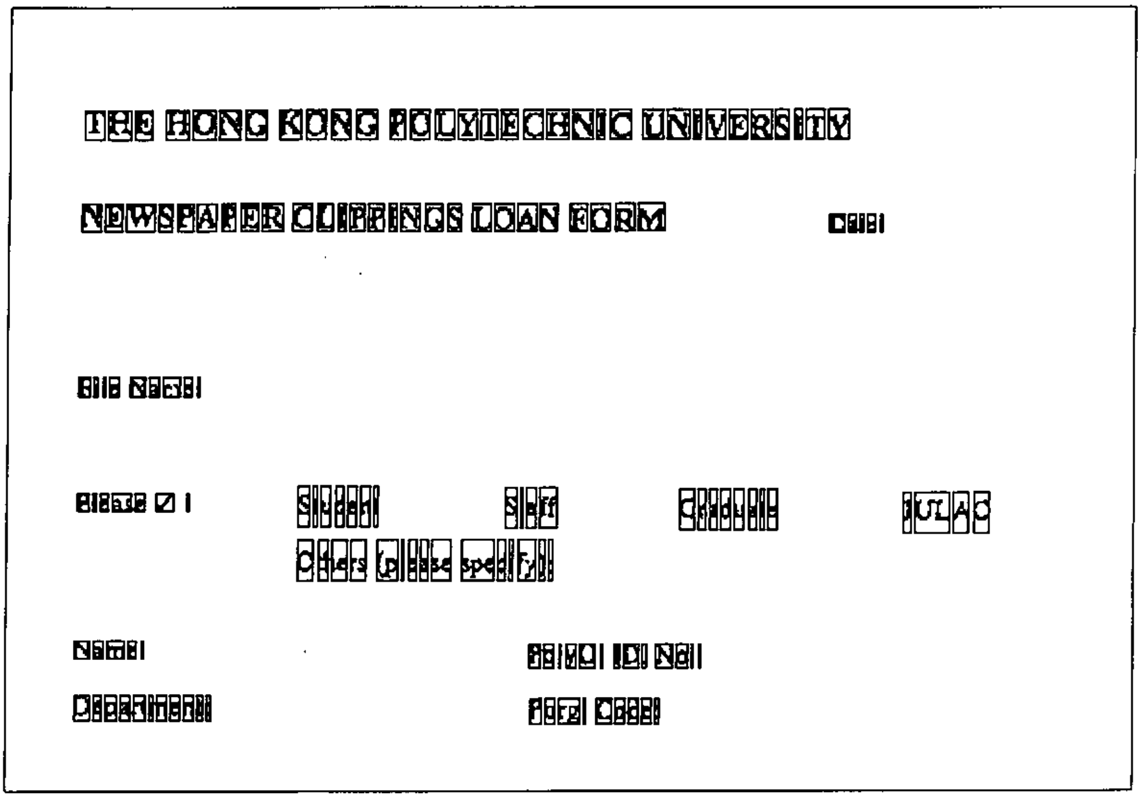

THE HONG KONG POLYTECHNIC UNIVERSITY
NEWSPAPER CLIPPINGS LOAN FORM

Fig. 5.3 An example of result of character segmentation.

5.3 Character Recognition Using Neural Network

In this project, a feedforward neural network with one hidden layer was employed as a classifier to recognize 64 English alphabets/digits/symbols. For those similar characters, such as C and c, we put them into a single category. As a result, we have 50 categories as shown in Table 5.1. Therefore, the number of output neurons is 50. A combined intensity features and skeleton-based features are used as the input.

Table 5.1 Categories of alphabets, digits and symbols

Class	Symbol	Class	Symbol	Class	Symbol	Class	Symbol
1	A	14	N	27	a	40	y
2	B	15	O, o, 0	28	b	41	2
3	C, c	16	P, p	29	d	42	3
4	D	17	Q	30	e	43	4
5	E	18	R	31	f	44	5
6	F	19	S, s	32	g	45	6
7	G	20	T	33	h	46	7
8	H	21	U, u	34	i	47	8
9	I, l, 1	22	V, v	35	j	48	9
10	J	23	W, w	36	n	49	(
11	K, k	24	X, x	37	q	50	)
12	L	25	Y	38	r		
13	M, m	26	Z, z	39	t		

5.3.1 Feature Extraction

Two types of features, intensity features and skeleton-based features, are used as input of the neural network classifier. To solve the problem of different sizes of different characters, each segmented binary image is normalized to a standard size of 64 × 64.

Intensity features are fast and easy to extracted. To reduce the number of intensity features, a normalized character image is further divided into 16 × 16 equal regions as shown in Fig. 5.4. The intensity of each region is used as a feature and thus, in total 256 features are obtained. These features reflect the intensity distribution of a character image.

It is believed that structure information is important in human recognition. To obtain the structural information of a character, skeleton-based features are extracted from the standard 64 × 64 character image.

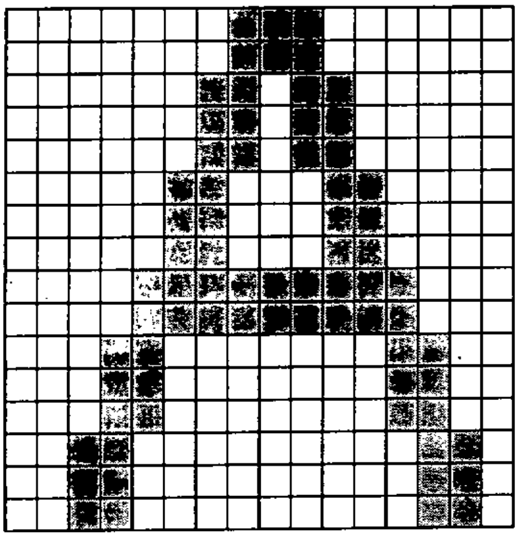
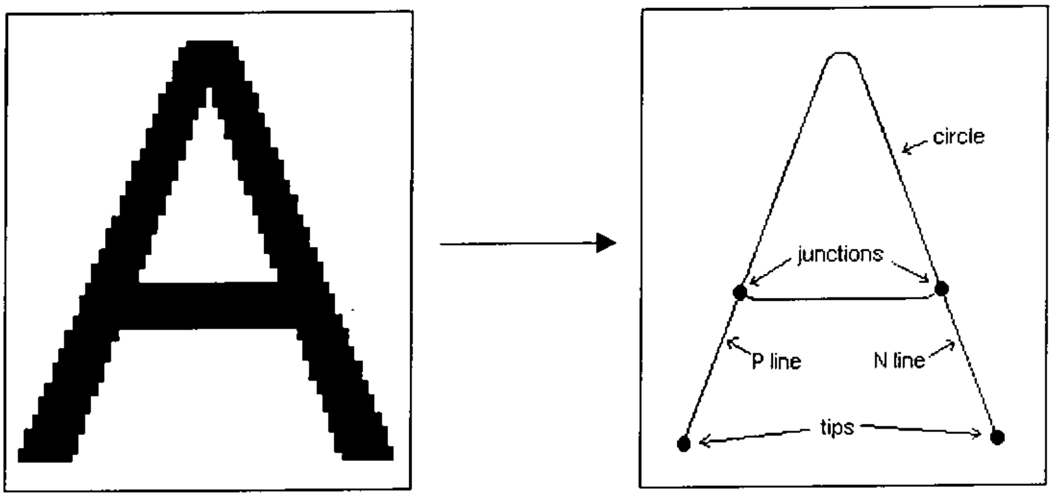


Fig. 5.4 Intensity features extraction of a character



(a) (b)

Fig. 5.5 An example of thinning and segment representation: (a) original image; (b) skeleton image.

To extract the skeleton-based features, a thinning algorithm is first applied to convert a binarized character image to a skeleton image as shown in Fig. 5.5(b). In a skeleton image, a set of junction points and tips (end points) are found and segments between these points are traced. On curve (but non-circular) segments, corner points are identified based on the rate of direction change. A labeling method is given below [31]:

- A node set in a skeleton image is defined as the collection of tips (points that have one neighbor), corners (points that have two neighbors and where an abrupt change of line direction occurs), and junctions (points that have more than two neighbors).
- A branch is a segment connecting a pair of adjacent nodes. A circle is a special branch connecting to a single node. Each segment is categorized as one of 12 types as shown in Figure 5.6, namely H line, V line, P line, N line, C curve, D curve, A curve, V curve, S curve, Z curve, curve, and circle. Type 'curve' is reserved for a curve segment which cannot be fitted into any of the other six curve types.
- The measure of straightness of a branch is determined by fitting a straight line using the least-square-error method. The straightness of a non-circular branch is defined as

  $$f_{SL} = \begin{cases} 1 - S / S_T & if \quad S < S_T \\ 0 & if \quad S \geq S_T \end{cases} \tag{5.2}$$

  where S_T is the threshold of the fitting error. A branch is classified as a curve, if $0 \leq f_{SL} < 0.5$, or a straight line, if $0.5 \leq f_{SL} < 1$
- According to its angle with the horizontal direction, a straight line segment is classified into H line, V line, P line or N line. A curve segment is categorized into one of six curve types based on its shape information. If a curve segment cannot be assigned to any of these types, it is considered to be 'curve'.

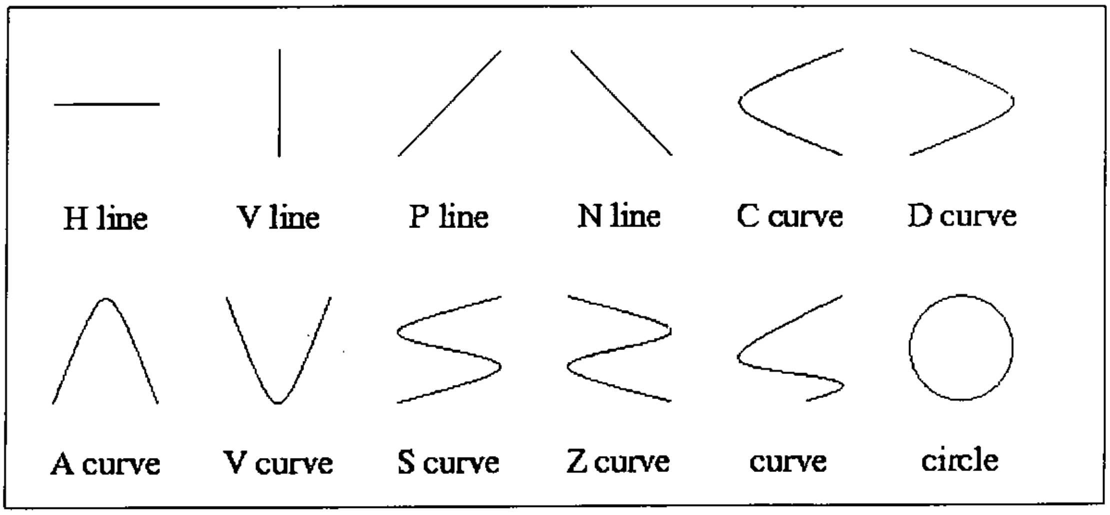


Fig. 5.6 Twelve types of line segments

Six longest segments (in the order of decreasing lengths) are considered for each skeleton image. Four features are associated with a segment. They are:

1. The type of a segment
2. the normalized length of the *i*-th segment
3. the normalized horizontal coordinate
4. the normalized vertical coordinate

Besides the four features for each of six segments, the number of segments that a character has is used as a feature, too. The other features used include the numbers of end points in each of four domains, the normalized total length, the center coordinates of a skeleton character image, the numbers of straight lines, curves and circles, and the aspect ratio of the image. Altogether 36 skeleton-based features are used. Together with the intensity features, in total, 292 features are extracted.

5.3.2 Neural Network Classifier

A two-layer (one hidden layer) neural network trained by conjugate gradient method is employed as a classifier. Since there are 292 features and 50 classes, 292 input nodes are used in the input layer and 50 nodes in output layer. And we use 150 nodes in the hidden layer.

Six sets of model characters were used for training. Thus, totally 450 sample characters were extracted for training. These characters were fed into the 292-150-50 neural network to train the neural network for recognition. Then we apply the recognition system on 4 form images which were scanned with resolution of 100 *dpi* and 6 form images which were scanned with a higher resolution of 150 *dpi*. It is found that the character recognition system can achieve an accuracy of 85% to 87.5% for the correctly segmented characters from the form images scanned with a low resolution of 100 *dpi* and a much higher accuracy, 94% to 96.6%, can be achieved if the resolution of the images increased from 100 *dpi* to 150 *dpi*. The recognition result of the form image showed in Fig. 5.3 with 100 *dpi* resolution is as following:

```
THEHONGKONGPOLYTECHNICUNIVERStTY
NEWSPAPERCLfPPINGSLOANFORM
DaIC:
riIgNaMe:
StNdent
Stee
GradMRte
PIeMeZt
I(AC
OeMMYICaWWfiN):
NaMe:
POIyU:IDMNOI:
DePRnInentI
POMtCOde:
```

The bolded characters were correctly segmented characters and the characters in italic were wrongly classified.

5.4 Concluding Remarks

English text fields from script determination must go through a character segmentation process. An algorithm based on vertical projection and peak-to-valley function is adopted to locate possible segmentation points.

A feedforward neural network with one hidden layer is employed to classify the segmented characters. In this character recognition system, two types of features, intensity features and skeleton-based features, are used as input of the neural network classifier. Each character image is first normalized to a standard size of 64 × 64. The normalized character is divided into 16 × 16 equal regions. The intensity of each region is used as a feature and thus, in total, 256 intensity features are obtained. To extract the skeleton-based features, a thinning algorithm is first applied to convert a binarized character to a skeleton image. Six longest segments (in the order of decreasing lengths) are extracted for each skeleton image. Four features are used to characterize a segment and 12 other features are also extracted from the skeleton image. Thus, altogether 36 skeleton-based features are extracted. Together with 256 intensity features, in total, 292 features are used for character recognition.

We made use of a 292-150-50 feedforward neural network to recognize 64 English alphabets/digits/symbols in 50 categories. A recognition rate of about 96% was achieved on document images scanned with 150 *dpi*. The performance can be further improved if the context information is also considered.

Chapter 6

Conclusion

6.1 General Conclusion

In this thesis, an automatic form reading system is developed. The system has two main parts, image preprocessing and form reading. The main task of image preprocessing is to locate the information fields in form images, which mainly includes image binarization, page segmentation, text and lines extraction, skew estimation and correction.

Image binarization: Combined histogram and a local contrast feature was used to binarize document images. An image-adaptive threshold was first determined from the intensity histogram of the image. A measure of local contrast of each pixel in the image was then computed to find out the relatively darker pixels in small sub-images. The binarization results were satisfactory for most of document images tested.

Page segmentation: The Run Length Smoothing Algorithm (RLSA) was first applied to the binary image to check the horizontal connection between black (foreground) pixels. The 8-neighborhood connection of the black pixels was then checked so that a connected component is formed. By this approach, the blocks that potentially contain useful information can be extracted even the image is skewed.

Text and lines extraction: Several criteria were used to classify the extracted blocks into text, lines and others. These criteria include the height and width of a block, the average horizontal length of the black runs in a block, and the number of vertical cuts of each vertical line in a block. Experimental results show that these criteria are quite reliable in classifying blocks.

Skew estimation and correction: Based on block and lines extraction, the skew angle of an image was determined by a majority vote on line orientations. This approach is computationally very efficient and effective for those forms containing a lot if lines. For forms without any line, a correction method can be adopted.

In form reading part, we have made investigations in script determination of Chinese and English textural images, fast keywords matching in images, and English character segmentation and recognition.

A neural network approach was applied for script determination. In this approach, two neural networks were employed. The first neural network was trained to derive a set of 15 masks that are used for extracting 15 features. The coefficients of masks were quantized to the nearest pre-defined values within [-2, -2] for reduced computational complexity. The second neural network of a smaller size was then trained with 15 extracted features as the input to perform the final script determination. Experimental results show that almost all of the test textual documents can be correctly classified.

In keywords matching, a Dynamic Recognition Neural Network (DRNN) was adopted. For English text, projection profiles (x and y) were used as features while for Chinese text, contour features were utilized. Experiments were carried out on twenty-nine name cards with chosen keywords, telephone and facsimile, of different formats (whole words or short form of the words). For English textual images, an accuracy of 91% to 92% can be achieved while for Chinese ones, 89% to 90% can be achieved.

English character segmentation and recognition was performed to the text fields that contain English text. Vertical projection and a peak-to-valley function were used for character segmentation. A feedforward neural network was then used for character recognition. Both intensity features and structure-based features extracted from the skeleton image were utilized. In total, 292 features were used. The recognition system can achieve an accuracy of about 86% for the correctly segmented characters with an image resolution of 100 *dpi* and a much higher accuracy of 96% with a resolution of 150 *dpi*.

6.2 Future Development

To make the system practical, a module for field identification and data extraction should be added (see Figure 1.1). The module should be able to differentiate the field words from the data filled in by the user. Separation of printed and handwritten characters may have to be performed.

In current system, some characters are recognized as groups. Context information has to be used to further identify these characters. For example, if the recognition result of consecutive four characters are 'N', 'a', 'M', 'e', we can use the rule that an uppercase letter only appears at the beginning of a word that contains small-case letters. Therefore, the result should read 'Name'.

Although the system developed in this thesis can generally handle different form images, a form registration module is still necessary for practical uses. The form registration module detects unique topological features of a form and retrieves the detailed description of a blank form stored in a form database which is the same form as the one to be processed. Since most of the offices usually deal with a limited number of forms, the registration module can increase both the speed and the accuracy of the system.

An algorithm should also be developed and added to the system to handle the forms with colored and/or patterned background. These types of forms are also commonly used in our daily life, such as bank checks and credit card application forms.

Finally, the system should have an input device, such as an optical scanner or a high-resolution digital camera. It should also support a function that allows users to make correction on the form reading results.

Appendix A: Implementation of a Prototypic Form Reading System

Components of the System

In our form reading system, we use a Hewlett Packard ScanJet IIcx desk-top scanner to scan forms. And we run the system on a PC with the following configurations:

- Pentium 166MHz CPU
- 32MB SDRAM
- Segate 2.1GB SCSI Harddisk × 2

A Linux OS with version 1.1.12 was installed on the PC.

Programming

The reading system is composed of many program modules in C. Here are the descriptions of the modules:

1. Read /Write access to an image file: This module is used for reading/writing of SUN raster image files. The reading part can read the header of an image file and retrieves the data for the form reading system. The writing part can create a file with a header of the same format and save the data into a file.
2. Binarization: It is composed of two parts, global thresholding and local contrast thresholding. The first part is used to obtain the intensity histogram of an image and to determine a threshold using the two highest peaks. The second part

computes the local contrast function as described in chapter 2 and determines a suitable threshold.

3. Page segmentation: This module performs the horizontal Run Length Smoothing Algorithm (RLSA) to connect neighboring components. The resultant images are used for block extraction by checking the 8-neighbourhood connections of the black pixels.
4. Text and lines classification: This module applies several criteria on the extracted blocks to identify the natures of data in the blocks.
5. Skew detection and correction: This module detects the skew angle of a document image based on the majority vote of line orientations and rotates the image accordingly.
6. Feature extraction for script determination: This module extracts square blocks from text lines and normalize the blocks to a standard size. Feature extraction is performed using the pre-trained 15 masks.
7. Feedforward neural network for script determination: A trained two-layer neural network is used to perform script determination of the extracted text blocks.
8. Dynamic Recognition Neural Network (DRNN): The trained DRNN is employed here so that if a test sample is inputted to this module, the corresponding classification result will be returned.
9. Character segmentation: This module computes the vertical projection function and the peak-to-valley function for each text block to perform separation of characters.
10. Feature extraction for character recognition: It is used for extracting the features for each segmented character.
11. Feedforward neural network for character recognition: Here, another trained neural network is used to perform the classification of characters by inputting the features of the characters extracted in Module 10.
12. User interface: The whole software package is run in a console in which the user can select the function he wants the system to perform and obtain the corresponding results. A third-party graphic viewer is used in this module.

13. Caching: This module saves the processing history to ensure every process in the system will not be repeated. The results of each process are also cached. It is used to save both the time and CPU loading.

Appendix B: Use of the Form Reading System

To use the form reading system that we developed, you have to have an IBM PC with Linux OS version 1.1.12 or above installed which must support X-Window and has the XV Viewer installed. Moreover, an image input device, such as an optical scanner is needed.

Before start

Make sure the Linux OS is running and start X-Window section. On an X-Window, open a shell for running the program since it has to be run in a command line. Use the image input device to get the required digitized image and store it as a gray-scaled image in SUN raster format (with an extension 'ras').

Run the program

To start running the program, type in the command line in the shell:

```
SHELL%>>[Program_name] [Image_filename]
```

where "SHELL%>>" is the prompt of the shell, [Program_name] is the name of the program which is "formread" in this thesis, and [Image_filename] is the filename of the required input image. For example:

```
\usr\local\bin>formread \home\root\form1.ras
```

After that, you will see the following messages in the shell:

```
Please select the options :
1. Obtain the binarized image.
2. Obtain the image of blocks extraction.
3. Obtain the image of skew correction.
4. Obtain the image of lines extraction.
5. Obtain the image of text block outlines.
6. Obtain the image of text extraction.
7. Obtain the image of character separation.
8. Obtain the result of character recognition.
9. Exit.

?
```

Type in the correspond number to get the required result. Except for the result of character recognition that will be displayed by a standard ‘vi’ editor, all the results will be displayed on an XV Viewer after the processes is finished. To quit the XV Viewer and continue running the program, you can press ‘q’ in the keyboard, or click the right mouse button on the XV Viewer and click the “QUIT” button in the control panel. For character recognition, you may press in sequence, ‘:’, ‘q’, and ‘<Enter>’ to quit the ‘vi’ editor.

Bibliography

[1] Y. Y. Tang, S-W. Lee and C. Y. Suen, "Automatic document processing: a survey", *Pattern Recognition*, **29**(12), 1931-1952 (1996)

[2] S. W. Lam, L. Javanbakht and S. N. Srihari "Anatomy of a Form Reader", *Proceedings of the Second International Conference on Document Analysis and Recognition*, 506-509 (1993)

[3] W. Chin, P. Gala , V.K. L. Huang, "Integrated imaging solution for industry and business", *Proceedings of the Second International Conference on System Integration (ICSI'92)*, 476-484, June 1992, Morristown, NJ, U.S.A.

[4] A. K. Jain and Z. Yu, "Page segmentation using texture analysis", *Pattern Recognition*, **29**, 743-770 (1996)

[5] G. Nagy, S. Seth and M. Viswanathan, "A prototype document image analysis system for technical journals", *IEEE Comput.*, **25**, 10-22 (July 1992)

[6] J. L. Fisher, S. C. Hinds and D. P. D'Amato, "A rule-based system for document image segmentation", *Proceeding s of the 10th International Conference on Pattern Recognition (ICPR)*, 567-572, June 1990, Atlantic City, New Jersey, U.S.A.

[7] H. Fujisawa, Y. Nakano and K. Kurino, "Segmentation methods for character recognition: form segmentation to document structure analysis", *Proceedings of The IEEE*, **80**(7), 1079-1092 (1992)

[8] S-W. Lee, D-J. Lee, H-S. Park, "A new methodology for gray-scale character segmentation and recognition", *IEEE Transaction on Pattern Analysis And Machine Intelligence*, **18**(10), 1045-1050, (1996)

[9] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development", *Proceedings of the IEEE*, **80**(7), 1029-1058 (1992)

[10] F. Cesarini, M. Gori, S. Marinai, and G. Soda, "INFORMys: a flexible invoice-like form-reader system", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, **20**, 730-745 (1998)

[11] H. Yan, "Skew correction of document images using interline cross-correlation", *CVGIP: Graphical Models And Image Processing*, **55**(6), 538-543 (1993)

[12] L. O'Gorman, "The document spectrum for page layout analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1162-1173 (1993)

[13] D. J. Ittner and H. S. Baird, "Language-free layout analysis", *Proceedings of Second International Conference on Document Analysis and Recognition*, 336-340 (1993)

[14] Y. Y. Tang, C. Y. Suen, C. D. Yan, and M. Cheriet, "Financial Document Processing Based on Staff Line and Description Language", IEEE Trans. System, Man, and Cybernetics, 25(5), 738-753 (1995)

[15] S. W. Lam, and S. N. Srihari, "Multi-domain document layout understanding", *Proceedings of First International Conference on Document Analysis and Recognition*, 112-120 (1991)

[16] J. Yuan, L. Xu, and C. Y. Suen, "Form Items Extraction by Model Matching", Proc. First Int'l Conf. Document Anal. Recog., 752-755 (1995)

[17] T. Watanabe, Q. Luo, and Sugie, "Layout recognition of multi-kinds of table-form documents", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(4), 432-445 (1995)

[18] R. Casey, D. Ferguson, K. Mohiuddin, and E. Walach, "Intelligent form processing system", *Machine Vision and Applications*, **5**(5) 143-155 (1992)

[19] Y. Liu and S. N. Srihart, "Document image binarization based on texture features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(5) 540-544 (1997)

[20] M-S. Chang, S-M. Kang, W-S. Rho, H-G. Kim, and D-J. Kim, "Improved binarization algorithm for document image by histogram and edge detection", *Proceedings of the Third International Conference on Document Analysis and Recognition*, **2**, 636-639 (1995)

[21] O. T. Akindele and A. Belaid, "Page segmentation by segment tracing", *Proceedings of Second International Conference on Document Analysis and Recognition*, 91-94 (1993)

[22] T. Pavlidis and J. Zhou, "Page segmentation and classification, " *CVGIP: Image Understanding*, **54**, 484-486 (1992)

[23] W. Postl, "Detection of linear oblique structures and skew scan in digitized documents", *Proc. of 8th Int. Conf. on Pattern Recognition*, 687-689 (1986)

[24] P. Parodi and G. Piccioli, "A fast and flexible statistical method for text extraction in document pages", *Proceedings on Computer Vision and Pattern Recognition (CVPR'96)*; 619-624 (1996)

[25] Y. Lu, B. Haist, L. Harmon, J. Trenkle and R. Vogt, "An accurate and efficient system for segmenting machine-printed text", *U.S. Postal Service 5th Advanced Technology Conference, Washington D.C.*, **3**, A-93-A-105 (1992)

[26] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(6), 910-918 (1988)

[27] S. Tsujimoto and H. Asada, "Resolving ambiguity in segmenting touching characters", *1st Internation Conference on Document Analysis and Recognition, Munich, Germany*, 1023-1026 (1991)

[28] Y. Tsuji and K. Asai, "Character image segmentation", *SPIE, Applications of Digital Image Processing VII*, **504**, 2-10 (1984)

[29] J. Hochberg, P. Kelly, T. Thomas and L. Kerns, "Automatic script identification from document images using cluster-based templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(2), 176-181 (1997)

[30] S. L. Wood, X. Yao, K. Krishnamurthi, and L. Dang, "Language identification for printed text independent of segmentation", *Proceedings of International Conference on Image Processing*, **3**, 428-431 (1995)

[31] Z. Chi, M. Sutes and H. Yan, "Handwritten digit recognition using combined ID3-derived fuzzy rules and Markov chains", *Pattern Recognition*, **29**(11), 1821-1833 (1996)

[32] G. Ciardiello, G. Scafuro, M. T. Degrandi, M. R. Spada, and M. P. Roccotelli, "An experimental system for office document handling and text recognition", *Proc. of 9th Int. Conf. on Pattern Recognition*, 739-743 (1988)

[33] Y. Yong, "Handwritten Chinese character recognition via neural networks", *Pattern Recognition Letters*, **7**, 19-25 (1988)

[34] M. C. K. Yang, J-S. Lee, C-C Lien, and C-L. Huang, "Hough transform modified by line connectivity and line thickness", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, **19**(8), 905-910 (1997)

[35] H. Kalviainen, P. Hirvonen, L. Xu and E. Oja, "Probabilistic and nonprobabilistic Hough Transform: overview and comparisons", *Image and Vision Computing*, **13**, 239-252 (1995)

[36] V. F. Leavers, "Survey – which Hough Transform", *Comput Vision, Graphics and Image Processing: Image Understanding*, **58**, 250-264 (1993)

[37] E. Giuliano, O. Paitra, and L. Stringa, "Electronic Character Reading System", U.S. Patent 4047,15,1977

[38] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis", *Computer Vision and Graphics Image Processing*, **20**, 327-352 (1989)

[39] G. Nagy and S. C. Seth, "Hierarchical representation of optical scanned documents", *Proceedings of the International Conference on Pattern Recognition*, 347-349 (1984)

[40] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents", *Computer Vision, Graphics and Image Processing*, **20**, 375-390 (1982)

[41] Y. Lu, "Machine printed character segmentation-an overview", *Pattern Recognition*, **28**(1) 67-80 (1995)

[42] A. Spitz, "Multilingual Document Recognition", Electronic Publishing, Document Manipulation, and Typography, R. Furuta, ed. Cambridge Univ. Press, 193-206 (1990)

[43] C. C. Chiang and H. C. Fu, "Using neural nets to recognize handwritten/printed characters", *Proceedings of Fifth Annual European Computer Conference on Advanced Computer Technology, Reliable Systems and Applications*, 492-496 (1991)

[44] M. M. Menon and K. G. Heinemann, "Classification of patterns using a self-organizing neural network, " *Neural Networks*, **1**, 201-215 (1988)

[45] K. Fukushima, "A neural network model for selective attention in visual pattern recognition", *Biological Cybernetics*, **55**, 5-15 (1986)

[46] J-Y. Yoo, M-K. Kim, S. Y. Han and Y-B. Kwon, "Line removal and restoration of handwritten characters on the form documents", *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 1, 128-131 (1997)

[47] J-S. Chen and D-C Tseng, "Overlapped-character separation and reconstruction for table-form documents", *Proceedings of the International Conference on Image Processing*, 233-236 (1996)

[48] Z. Chi and H. Yan, "Image segmentation using fuzzy rules derived from K-means clusters", *Journal of Electronic Imaging*, **4**(2), 199-206 (1995)

[49] R. Kasturi, S. T. Bow, W. EL-Masri, J. Shan, J. R. Gattiker, and U. B. Mokate, "A system for interpretation of line drawings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(10), 978-992 (1990)

[50] R. M. Haralick, "Document image understanding: geometric and logical layout", *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition (CVPR)*, 385-390 (1994)

[51] J. Illingworth and J. Kittler, "A survey of the Hough Transform", *Comput. Vision Graphics Image Processing*, **44**, 87-116 (1988)

[52] T. Tsunoda, T. Shiraishi and H. Tanaka, "Character Recognition ny Associative Completion on Words", *Proceedings of 1993 International Joint Conference on Neural Networks*, 1135-1138 (1993)

[53] S. Kuo and O. E. Agazzi, "Visual Keyword Recognition Using Hidden Markov Models", *Proceedings of the International Conference on Pattern Recognition*, 329-334 (1993)

[54] F. R. Chen, L. D. Wilcox and D. S. Bloomberg, "Word Spotting in Scanned Images Using Markov Models", *Proceedings of the International Conference on Pattern Recognition*, V1-V4 (1993)

[55] L-P. Zhang, L-M Li, and Z. Chi, "An on-line adaptive neural network for speech recognition", *International Journal of Speech Technology*, **2**, 241-248 (1998)

[56] Y. Kurosawa and H. Asada, "Attributed string matching with statistical constraints for character recognition", *Proc. Eighth Int. Conf. Pattern Recognition, Paris, France*, 1063-1067 (1986)

[57] P. Siy and C. S. Chen, "Fuzzy logic for handwritten numeral character recognition", *IEEE Trans. on Syst.*, 570-574 (1974)

[58] S. Kahan and T. Pavlidis, “On the recognition of printed characters of any font and size”, *IEEE Trans. Pattern Anal. Mach. Intell.*, **PAMI-9**, 274-287 (1987)