

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Preprocessing Techniques in Character Recognition

Yasser Alginahi
Taibah University
Kingdom of Saudi Arabia

1. Introduction

The advancements in pattern recognition has accelerated recently due to the many emerging applications which are not only challenging, but also computationally more demanding, such evident in Optical Character Recognition (OCR), Document Classification, Computer Vision, Data Mining, Shape Recognition, and Biometric Authentication, for instance. The area of OCR is becoming an integral part of document scanners, and is used in many applications such as postal processing, script recognition, banking, security (i.e. passport authentication) and language identification. The research in this area has been ongoing for over half a century and the outcomes have been astounding with successful recognition rates for printed characters exceeding 99%, with significant improvements in performance for handwritten cursive character recognition where recognition rates have exceeded the 90% mark. Nowadays, many organizations are depending on OCR systems to eliminate the human interactions for better performance and efficiency.

The field of pattern recognition is a multidisciplinary field which forms the foundation of other fields, as for instance, Image Processing, Machine Vision, and Artificial Intelligence. Therefore, OCR cannot be applied without the help of Image Processing and/or Artificial Intelligence. Any OCR system goes through numerous phases including: data acquisition, preprocessing, feature extraction, classification and post-processing where the most crucial aspect is the preprocessing which is necessary to modify the data either to correct deficiencies in the data acquisition process due to limitations of the capturing device sensor, or to prepare the data for subsequent activities later in the description or classification stage. Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Hence, preprocessing is the preliminary step which transforms the data into a format that will be more easily and effectively processed. Therefore, the main task in preprocessing the captured data is to decrease the variation that causes a reduction in the recognition rate and increases the complexities, as for example, preprocessing of the input raw stroke of characters is crucial for the success of efficient character recognition systems. Thus, preprocessing is an essential stage prior to feature extraction since it controls the suitability of the results for the successive stages. The stages in a pattern recognition system are in a pipeline fashion meaning that each stage depends on the success of the previous stage in order to produce optimal/valid results. However, it is

evident that the most appropriate feature vectors for the classification stage will only be produced with the facilitation from the preprocessing stage. The main objective of the preprocessing stage is to normalize and remove variations that would otherwise complicate the classification and reduce the recognition rate.

2. Factors affecting character recognition quality

There are a number of factors that affect the accuracy of text recognized through OCR. These factors include: scanner quality, scan resolution, type of printed documents (laser printer or photocopied), paper quality, fonts used in the text, linguistic complexities, and dictionary used. “Foxing” and “text show through” found in old paper documents, watermarks and non-uniform illumination are examples of problems that affect the accuracy of OCR compared to a clean text on a white background. For example, Fig.1 (a) shows a grey-level document image with poor illumination and Fig.1 (b) shows a mixed content document image with complex background. Other factors include features of printing such as uniformity, text alignment and arrangement on the page, graphics and picture content (Tanner, 2004).

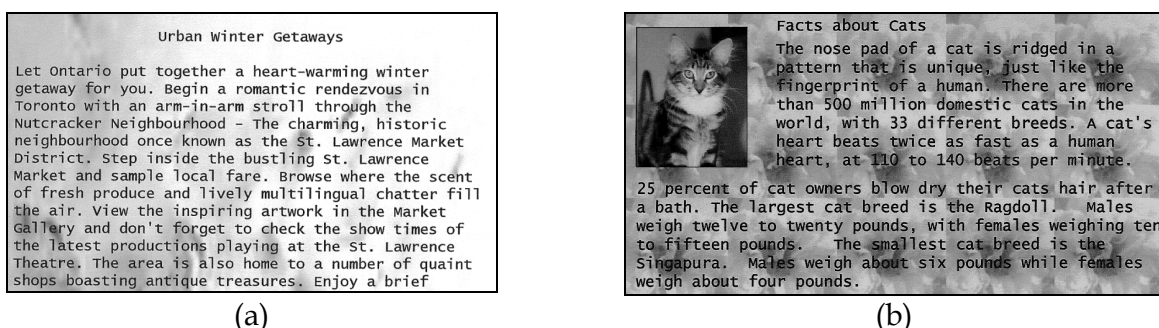


Fig. 1. Examples of document images with non-uniform/complex backgrounds

3. Importance of preprocessing in character recognition

The importance of the preprocessing stage of a character recognition system lies in its ability to remedy some of the problems that may occur due to some of the factors presented in section 2 above. Thus, the use of preprocessing techniques may enhance a document image preparing it for the next stage in a character recognition system. In order to achieve higher recognition rates, it is essential to have an effective preprocessing stage, therefore; using effective preprocessing algorithms makes the OCR system more robust mainly through accurate image enhancement, noise removal, image thresholding, skew detection/correction, page segmentation, character segmentation, character normalization and morphological techniques.

4. Preprocessing techniques

Preprocessing techniques are needed on colour, grey-level or binary document images containing text and/or graphics. In character recognition systems most of the applications use grey or binary images since processing colour images is computationally high. Such images may also contain non-uniform background and/or watermarks making it difficult to

extract the document text from the image without performing some kind of preprocessing, therefore; the desired result from preprocessing is a binary image containing text only. Thus, to achieve this, several steps are needed, first, some image enhancement techniques to remove noise or correct the contrast in the image, second, thresholding to remove the background containing any scenes, watermarks and/or noise, third, page segmentation to separate graphics from text, fourth, character segmentation to separate characters from each other and, finally, morphological processing to enhance the characters in cases where thresholding and/or other preprocessing techniques eroded parts of the characters or added pixels to them. The above techniques present few of those which may be used in character recognition systems and in some applications; few or some of these techniques or others may be used at different stages of the OCR system. The rest of the chapter will present some of the techniques used during the preprocessing stage of a character recognition system.

4.1 Image enhancement techniques

Image enhancement improves the quality of images for human perception by removing noise, reducing blurring, increasing contrast and providing more detail. This section will provide some of the techniques used in image enhancement.

4.1.1 Spatial image filtering operations

In image processing, filters are mainly used to suppress either the high frequencies in the image, *i.e.* smoothing the image, or the low frequencies, *i.e.* enhancing or detecting edges in the image. Image restoration and enhancement techniques are described in both the spatial domain and frequency domain, *i.e.* Fourier transforms. However, Fourier transforms require substantial computations, and in some cases are not worth the effort. Multiplication in the frequency domain corresponds to convolution in the time and the spatial domain. Using a small convolution mask, such as 3x3, and convolving this mask over an image is much easier and faster than performing Fourier transforms and multiplication; therefore, only spatial filtering techniques will be presented in this chapter.

Images captured often may be influenced by noise; however, the resulting images may not provide desired images for analysis. In addition, in images with acceptable quality, certain regions may need to be emphasized or highlighted. Spatial processing is classified into point processing and mask processing. Point processing involves the transformation of individual pixels independently of other pixels in the image. These simple operations are typically used to correct for defects in image acquisition hardware, for example to compensate for under/over exposed images. On the other hand, in mask processing, the pixel with its neighbourhood of pixels in a square or circle mask are involved in generating the pixel at (x, y) coordinates in the enhanced image.

4.1.1.1 Point processing

Point processing modifies the values of the pixels in the original image to create the values of the corresponding pixels in the enhanced image this is expressed in equation (1).

$$O(x,y) = T[I(x,y)] \quad (1)$$

Where, $I(x, y)$ is the original (input) image, $O(x, y)$ is the enhanced image and T describes the transformation between the two images. Some of the point processing techniques include: contrast stretching, global thresholding, histogram equalisation, log transformations and power law transformations. Some mask processing techniques include averaging filters, sharpening filters, local thresholding... etc.

4.1.1.1.1 Contrast stretching

The level of contrast in an image may vary due to poor illumination or improper setting in the acquisition sensor device. Therefore, there is a need to manipulate the contrast of an image in order to compensate for difficulties in image acquisition. The idea is to modify the dynamic range of the grey-levels in the image. A technique that could work in this case is called linear mapping, equation (2), to stretch the pixel values of a low-contrast image or high-contrast image by extending the dynamic range across the whole image spectrum from $0 - (L-1)$.

$$O(x,y) = O_1 + \left(\frac{O_2 - O_1}{I_2 - I_1} \right) [I(x,y) - I_1] \quad (2)$$

where O_1 corresponds to 0 and O_2 corresponds to the number of desired levels which is $(L-1 = 255)$. I_1 and I_2 provide the minimum and maximum values of the input grey-level range. The simplest form of processing is to adjust the brightness of an image by adding a bias value, b , to all the pixel values of an image; where $b > 0$ would increase the brightness of an image and $b < 0$ would darken the image. Also, a gain factor, a , may be used instead of a bias, where the product of a with the input pixel values modify the brightness of the output image. Values of $0 < a < 1$ will produce a darker image and values of $a > 1$ will produce a brighter image. Combining both bias and gain produces equation (3).

$$O(x, y) = a * I(x, y) + b \quad (3)$$

In this case, we need to specify both the gain and bias values, but in practicality it may be difficult to do so; therefore, the solution would be to map the input image range (I_1, I_2) to the output image range (O_1, O_2) where O_1 corresponds to 0 and O_2 corresponds to the number of desired levels, hence linear mapping defined in equation (2).

4.1.1.1.2 Global image thresholding

Image thresholding is the process of separating the information (objects) of an image from its background, hence, thresholding is usually applied to grey-level or colour document scanned images. Thresholding can be categorised into two main categories: global and local. Global thresholding methods choose one threshold value for the entire document image, which is often based on the estimation of the background level from the intensity histogram of the image; hence, it is considered a point processing operation. On the other hand, local adaptive thresholding uses different values for each pixel according to the local area information. There are hundreds of thresholding algorithms which have been published in the literature and presenting all methods would need several books, therefore, the purpose here is to present some of the well-known methods.

Global thresholding methods are used to automatically reduce a grey-level image to a binary image. The images applied to such methods are assumed to have two classes of pixels (foreground and background). The purpose of a global thresholding method is to automatically specify a threshold value, T , where the pixel values below it are considered foreground and the values above are background. A simple method would be to choose the mean or median value of all the pixels in the input image, the mean or median will work well as the threshold, however, this will generally not be the case especially if the pixels are not uniformly distributed in an image. A more sophisticated approach might be to create a histogram of the image pixel intensities and use the valley point (minimum) as the threshold. The histogram approach assumes that there is some average value for the background and object pixels, but that the actual pixel values have some variation around these average values. However, this may be computationally expensive, and image histograms may not have clearly defined valley points, often making the selection of an accurate threshold difficult. One method that is relatively simple and does not require much specific knowledge of the image is the iterative method (Gonzalez, et al., 2004) which is explained below.

The iterative procedure is

Step 1: Select an initial threshold value (T), randomly or according to any other method desired such as the mean or median value of the pixels in the image.

Step 2: Segment the image, using T , into object and background pixels. R_1 (background region) consists of pixels with intensity values $\geq T$ and R_2 (objects region) consists of pixels with intensity $< T$.

Step 3: Calculate the average of each region, μ_1 and μ_2 for regions R_1 and R_2 , respectively.

Step 4: Compute the new threshold value T as given in equation (4).

$$T = 1/2(\mu_1 + \mu_2) \quad (4)$$

Step 5: Repeat the steps from 2 – 4 using the new T until the new threshold matches the one before it.

In the literature, many thresholding methods have been published, for example, Sahoo et al. compared the performance of more than 20 global thresholding algorithms using uniformly or shape measures. The comparison showed that Otsu class separability method gave best performance (Sahoo et al., 1988; Otsu, 1979). On the other hand, in an evaluation for change detection by Rosin & Ioannidis concluded that the Otsu algorithm performed very poorly compared to other global methods (Rosin & Ioannidis, 2003, Otsu, 1979). The OCR goal-directed evaluation study by Trier and Jain examined four global techniques showing that the Otsu method outperformed the other methods investigated in the study (Trier & Jain, 1995). In addition, Fischer compared 15 global methods and confirmed that the Otsu method is preferred in document image processing (Fischer, 2000). The Otsu method is one of the widely used techniques used to convert a grey-level image into a binary image then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

The Otsu method searches for the threshold that minimises the intra-class variance, defined in equation (5) as a weighted sum of variances of the two classes (Otsu, 1979).

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (5)$$

weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 is the variance of these classes. Otsu shows that minimising the intra-class variance is the same as maximising inter-class variance, equation (6):

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (6)$$

this is expressed in terms of class probabilities ω_i and class means μ_i which in turn can be updated iteratively.

The algorithm steps are:

- Compute the histogram and probabilities of each intensity level
- Initialize $\omega_i(0)$ and $\mu_i(0)$
- Step through all threshold values $t = 1 \dots$ to maximum intensity.
 - Update $\omega_i(0)$ and $\mu_i(0)$
 - Compute the maximum $\sigma_b^2(t)$, which corresponds to the desired threshold.

4.1.1.1.3 Histogram processing

Histogram processing is used in image enhancement and can be useful in image compression and segmentation processing. A histogram simply plots the frequency at which each grey-level occurs from 0 (black) to 255 (white). Scanned or captured images may have a limited range of colours, or are lacking contrast (details). Enhancing the image by histogram processing can allow for improved detail, but can also aid other machine vision operations, such as segmentation. Thus, histogram processing should be the initial step in preprocessing. Histogram equalisation and histogram specification (matching) are two methods widely used to modify the histogram of an image to produce a much better image.

4.1.1.1.3.1 Histogram equalisation

Histogram equalisation is considered a global technique. It stretches the histogram across the entire spectrum of pixels (0 – 255). It increases the contrast of images for the finality of human inspection and can be applied to normalize illumination variations in image understanding problems. This process is quite simple and for each brightness level j in the original image, the new pixel level value (k) is calculated as given in equation (7).

$$k = \sum_{i=0}^j N_i / T \quad (7)$$

where the sum counts the number of pixels in the image (by integrating the histogram) with brightness equal to or less than j , and T is the total number of pixels (Russ, 2007). In addition, histogram equalisation is one of the operations that can be applied to obtain new images based on histogram specification or modification.

4.1.1.1.3.2 Histogram specification (Matching)

Histogram matching is a method in image processing of colour adjustment of two images using their image histograms.

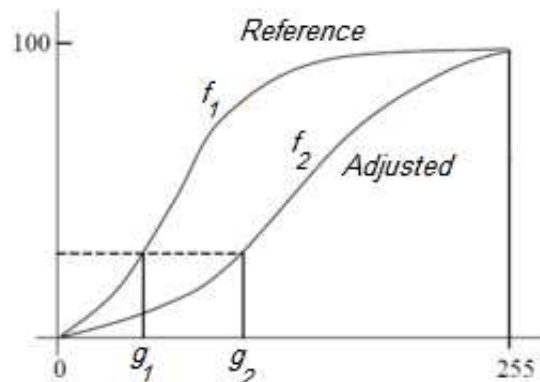


Fig. 2. Cumulative distributive functions for reference and adjusted images.

Histogram modification is the matching of the cumulative function f_2 of the image to be adjusted to the Cumulative Distribution Function (CDF) of the reference image f_1 . Histogram modification is done by first computing the histograms of both images then the CDFs of both the reference (f_1) and to be adjusted (f_2) images are calculated. This output of the histogram matching is obtained by matching the closest CDF f_2 to the reference image CDF f_1 . Then for each grey-level g_1 the grey-level g_2 is calculated for which $f_1(g_1) = f_2(g_2)$ as shown in Fig. 2, and this is the result of histogram matching function $M(g_1) = g_2$ (Horn & Woodham, 1979).

4.1.1.1.4 Log transformations

The general form of the log transformation is equation (8).

$$s = c \log (1 + r) \quad (8)$$

where c is a constant and it is assumed that $r \geq 0$. This transformation maps a narrow range of low grey-level values in the input image into a wider range of output levels and vice versa (Gonzalez et al., 2004).

4.1.1.1.5 Power law transformation

Power-law transformations have the general form shown in equation (9).

$$s = c(r + \varepsilon)^\gamma \quad (9)$$

where c and γ are positive constants and ε is an offset which is usually ignored since it is due to display calibration. Therefore; $s = c \cdot r^\gamma$, where values of $0 < \gamma < 1$ map a narrow range of dark input values into a wider range of output values, with the opposite being true for values of γ greater than 1. This shows that the power-law transformations are much more versatile in such application than the log transformation. However, the log function has the important characteristic that it compresses the dynamic range of images with large variations in pixel values. Due to the variety of devices used for image capture, printing,

and display respond according to the power law exponent, gamma, (γ), this factor needs to be corrected, thus power-law response phenomena or gamma correction which is given by $s = c \cdot r^{1/\gamma}$ (Gonzalez et al., 2004).

4.1.1.2 Mask processing

In mask processing, a pixel value is computed from the pixel value in the original image and the values of pixels in its vicinity. It is a more costly operation than simple point processing, but more powerful. The application of a mask to an input image produces an output image of the same size as the input.

4.1.1.2.1 Smoothing (Low-pass) filters

Average or mean filter is a simple, intuitive and easy to implement method of *smoothing* images, *i.e.* reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images. In general, the mean filter acts as a low-pass frequency filter and, therefore, reduces the spatial intensity derivatives present in the image. The idea of mean filtering is simply to replace each pixel value in an image with the mean ('average') value of its neighbours, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighbourhood to be sampled when calculating the mean. Often a 3×3 square kernel/mask is used, as shown in Fig. 3, although larger masks can be used (*e.g.* 5×5, 7×7, 9×9 ...) for more severe smoothing. Note that, a small kernel can be applied more than once in order to produce a similar, but not identical, effect as a single pass with a larger kernel. Also, the elements of the mask must be positive and hence the size of the mask determines the degree of smoothing. Therefore, the larger the window size used a blurring effect is produced causing small objects to merge with the background of the image (Nixon & Aguado, 2008).

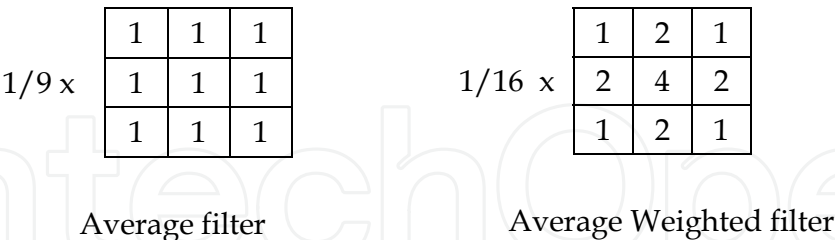


Fig. 3. 3×3 averaging kernels used in average filter.

The center coefficient of the mask is very important and other pixels are inversely weighted as a function of their distance from the center of the mask. The basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.

4.1.1.2.2 Sharpening (High-pass) filter

A sharpening filter is used to emphasize the fine details of an image (i.e., provides the opposite effect of smoothing). The points of high contrast can be detected by computing intensity differences in local image regions. The weights of the mask are both positive and negative. When the mask is over an area of constant or slowly varying grey-level, the result of convolution will be close to zero. When grey-level is varying rapidly within the neighbourhood, the result of convolution will be a large number. Typically, such points form the border between different objects or scene parts (i.e. edge). An example of a sharpening filter is the Laplacian filter which is defined in equation (10) below.

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y) \tag{10}$$

This implementation can be applied at all points (x,y) in an image by convolving the image with the following spatial mask Fig. 4(a) with an alternative definition of the digital second derivatives which takes into account the diagonal elements and can be implemented by the mask in Fig. 4(b).

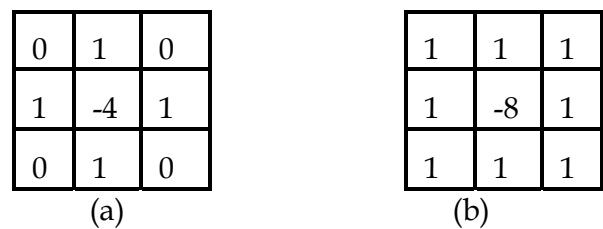


Fig. 4. 3x3 Laplacian filter masks

The Laplacian filter is a derivative operator which sharpens the image, but drives constant areas to zero; therefore, adding the original image back restores the grey-level tonality, equation (11).

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)] \tag{11}$$

Where, $f(x,y)$ is the input image, $g(x,y)$ is the output image and c is 1 if the centre coefficient of the mask is positive, or -1 if it is negative (Gonzales and Woods, 2002).

4.1.1.2.3 Median filter

A commonly used non-linear operator is the median, a special type of low-pass filter. The median filter takes an area of an image (3x3, 5x5, 7x7, etc.), sorts out all the pixel values in that area, and replaces the center pixel with the median value. The median filter does not require convolution. (If the neighbourhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) Fig. 5 illustrates an example of how the median filter is calculated. The median filter is effective for removing impulse noise such as “salt and pepper noise” which is random occurrences of black and white pixels.

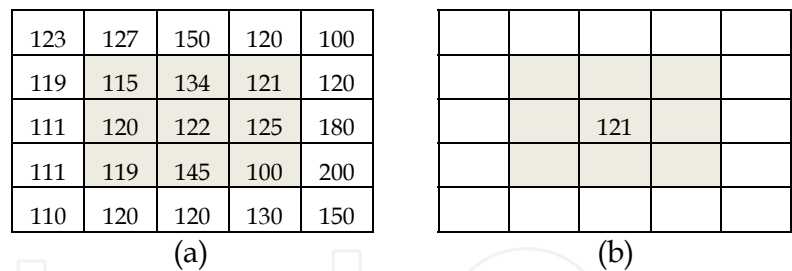


Fig. 5. (a) Input image (b) Filtered image using median filter showing only the centre pixel.

The sorted pixel values of the shaded area are: (100, 115, 119, 120, 121, 122, 125, 134 and 145), providing a median value of 121 in the output image.

4.1.1.2.4 Maximum filter

The maximum filter is defined as the maximum of all pixels within a local region of an image. The maximum filter is typically applied to an image to remove negative outlier noise. For the example in Fig. 5 the center pixel will take the maximum value 145.

4.1.1.2.5 Minimum filter

The minimum filter enhances dark values in the image; therefore, the darkest pixel then becomes the new pixel value at the centre of the window. For the example in Fig. 5 the centre pixel will be replaced by the minimum value of 100.

4.1.1.2.6 Range filter

The range filter is defined as the difference between the maximum and minimum pixel values within the neighbourhood of a pixel. For the example in Fig. 5 the centre pixel will be replaced by 45.

4.1.1.2 Local thresholding

Local thresholding techniques are used with document images having non-uniform background illumination or complex backgrounds, such as watermarks found in security documents if the global thresholding methods fail to separate the foreground from the background. This is due to the fact that the histogram of such images provides more than two peaks making it difficult for a global thresholding technique to separate the objects from the background, thus; local thresholding methods are the solution. The local thresholding techniques developed in the literature are mainly for specific applications and most of the time they do not perform well in different applications. The results could be over thresholding or under thresholding depending on the contrast and illumination. From the literature, several surveys have compared different thresholding techniques. The work of Trier and Jain evaluated the performance of 11 well-established locally adaptive binarisation methods (Trier & Jain, 1995). These techniques were compared using a criterion based on the ability of an OCR module to recognize handwritten numerals from hydrographical images. In this evaluation, the Niblack’s method, (Niblack, 1986), appears to be the best. This observation was applied for a specific application on certain hydro-graphic images using an OCR system. However, as concluded by the authors, if different sets of images used with different feature extraction methods and classifiers, then this observation may not

be accurate and another method could outperform the Niblack's method (Trier & Jain, 1995). The Niblack's method calculates the threshold by shifting a window across the image, and use local mean, μ , and standard deviation, σ , for each center pixel in the window. The threshold value for a pixel within fixed neighbourhood is a linear function of the mean and standard deviation of the neighbourhood pixels, with a constant gradient of $T(x, y)$, which is highly tunable, to separate objects well. Then the threshold is equation (12).

$$T(x, y) = \mu(x, y) + k \sigma(x, y) \quad (12)$$

The size of the neighbourhood should be small enough to serve local details, but at the same time large enough to suppress noise. The value of k is used to adjust how much of the total print object boundary is taken as a part of the given object. There have been several methods which introduced modifications to the Niblack's method, such as the work of Zhang and Tan who proposed an improved version of the Niblack's algorithm (Zhang and Tan, 2001). In addition, too many other thresholding methods based on different properties of the image were also developed. For example, the local thresholding method developed by Alginahi, uses the MLP-NN to classify pixels as background or foreground using statistical texture features to characterize the set of neighbourhood values of pixels related to its moments and measures of properties such as smoothness, uniformity and variability (Alginahi, 2004, 2008). In this work, five features were extracted from a window size 3x3 these are the centre pixel value of the window, mean, standard variation, skewness and entropy. These features were extracted from each pixel and its neighbourhood in the image and then passed into a MLP-NN to classify pixels into background (white) and foreground (black). The MLP-NN thresholding method proved to provide excellent results in thresholding documents with bad illumination, containing complex background and/or non-uniform background, such as those found in security documents. The MLP-NN thresholding method is a non-application specific and can work with any application provided that sufficient training is carried out.

4.2 Noise removal

The advancements in technology produced image acquisition devices with better improvements. While modern technology has made it possible to reduce the noise levels associated with various electro-optical devices to almost negligible levels, there are still some noise sources which cannot be eliminated. Images acquired through modern sensors may be contaminated by a variety of noise sources. By noise we refer to stochastic variations as opposed to deterministic distortions, such as shading or lack of focus. There are different types of noise that are related to the electronic capturing devices or the light source used such types of noise are photon, thermal, On-Chip electronic and quantisation. Most of the noise may be eliminated by the capturing sensors or the CCD cameras.

Document analysis systems benefit from the reduction of noise in the preprocessing stage this can provide a substantial improvement in the reliability and robustness of the feature extraction and recognition stages of the OCR system. A common manifestation of noise in binary images takes the form of isolated pixels, salt-and-pepper noise or speckle noise, thus; the processing of removing this type of noise is called filling, where each isolated pixel salt-and-pepper "island" is filled in by the surrounding "sea" (O'Gorman, et al., 2008). In grey-level images or median filters and low-pass filters such as average or Gaussian blur filters proved to eliminate isolated pixel noise. Gaussian blur and average filters are a better choice

to provide smooth texture to the image. On the other hand, periodic noise which manifests itself as impulse-like bursts which often are visible in the Fourier spectrum can be filtered using notch filtering. The transfer function of a Butterworth notch filter of order n , $H(u, v)$, is given by equation (13).

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u, v)D_2(u, v)} \right]^n} \quad (13)$$

Where

$$D_1(u, v) = [(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2]^{1/2} \quad (14)$$

And

$$D_2(u, v) = [(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2]^{1/2} \quad (15)$$

where (μ_0, v_0) and by symmetry $(-\mu_0, -v_0)$ are the locations of the notches and D is their radius, equations 14 - 15. The filter is specified with respect to the centre of the frequency rectangle. (Gonzalez et al., 2004).

4.3 Skew detection/correction

Due to the possibility of rotation of the input image and the sensitivity of many document image analysis methods to rotation of the image, document skew should be corrected. Skew detection techniques can be roughly classified into the following groups: analysis of projection profile, Hough transform, connected components, clustering, and Correlation between lines techniques. The survey by Hull and Taylor, investigated twenty-five different methods for document image skew detection. The methods include approaches based on Hough Transform analysis, projection profile, feature point distribution and orientation-sensitive feature analysis. The survey concluded that most of the techniques reported a range of up to 0.1 degrees accuracy, evidencing a strong need for further work in this area to help show the strengths and weaknesses of individual algorithms (Hull & Taylor, 1998). In addition, there are new techniques emerging for specific applications such as the method of Al-Shatnawi and Omar which is based on the center of gravity for dealing with Arabic document images (Al-Shatnawi & Omar, 2009). Therefore, the choice of using a skew detection/correction technique depends on the application and the type of images used.

4.4 Page segmentation

After image enhancement, noise removal and/or skew detection/correction, the next step in mixed content images or composite images is to perform page segmentation in order to separate text from halftone images, lines, and graphs. The result of interest should be an image with only text; therefore, document/page segmentation. Document segmentation can be classified into three broad categories: top-down, bottom-up and hybrid techniques. The top-down methods recursively segment large regions in a document into smaller sub-

regions. The segmentation stops when some criterion is met and the ranges obtained at that stage constitute the final segmentation results. On the other hand, the bottom-up methods start by grouping pixels of interest and merging them into larger blocks or connected components, such as characters which are then clustered into words, lines or blocks of text. The hybrid methods are the combination of both top-down and bottom-up strategies.

The Run-Length Smearing Algorithm (RLSA) is one of the most widely used top-down algorithms. It is used on binary images (setting 1 for white pixels and 0 for black pixels), by linking together the neighbouring black pixels that are within a certain threshold. This method is applied row-by-row and column-by-column, then both results are combined in a logical OR operation and finally a smoothing threshold is used to produce the final segmentation result. From the RLSA results, black blocks of text lines and images are produced. Finally a statistical classifier is used to classify these blocks (Wahl et al., 1982).

An example of bottom-up algorithm is the recursive X-Y method, which is also known as the projection profile cuts, it assumes documents are presented in a form of a tree of nested rectangular blocks (Nagy & Stoddard, 1986). Although the recursive X-Y cuts could decompose a document image into a set of rectangular blocks no details were given on how to define cuts. On the other hand, an example of a hybrid method is the segmentation approach of Kruatrachue and Suthaphan which consists of two steps, a top down block extraction method followed by a bottom-up multi-column block detection and segmentation method (Kruatrachue & Suthaphan, 2001). The segmentation is based on blocks of columns extracted by a modified edge following algorithm, which uses a window of 32×32 pixel so that a paragraph can be extracted instead of a character.

The above are only a few examples and hundreds of methods developed for document layout segmentation. To ensure the performance of most of these algorithms, a skew detection and correction algorithm is required in the preprocessing stage. In literature, the surveys in (Mao et al., 2003) and (Tang et al., 1996) provide detailed explanation on document analysis and layout representation algorithms. Most of the techniques explained are time consuming and are not effective for processing documents with high geometrical complexity. Specifically, the top-down approach can process only simple documents, which have specific format or contain some a priori information about the document. It fails to process the documents that have complicated geometric structures. The research in this area concentrates on binary images and grey images with uniform backgrounds. The images used were mainly scanned from technical journals and magazines that usually have specific formats. Document segmentation on grey-level images with complex or non-uniform backgrounds have not been widely investigated due to the complications in thresholding these images. Therefore, techniques are mainly geared to specific applications with specific formats and they tend to fail when specific parameters do not match. Alginahi, et al. used a local MLP-NN threshold to threshold images with uniform background and applied the RLSA with modified parameters to segment a mixed content document image into text, lines, halftone images and graphics (Alginahi et al., 2005,2008).

4.5 Character segmentation

Character segmentation is considered one of the main steps in preprocessing especially in cursive scripts such as Arabic, Urdu and other scripts where characters are connected together. Therefore, there are many techniques developed for character segmentation and most of them are script specific and may not work with other scripts. Even in printed

handwritten documents, character segmentation is required due to touching of characters when written by hand. For example, printed Latin characters are easy to segment using horizontal and vertical histogram profiles; however, smaller fonts and those containing serifs may introduce touching which will need further processing to solve the touching problem.

4.6 Image size normalization

The result from the character segmentation stage provides isolated characters which are ready to be passed into the feature extraction stage; therefore, the isolated characters are normalized into a specific size, decided empirically or experimentally depending on the application and the feature extraction or classification techniques used, then features are extracted from all characters with the same size in order to provide data uniformity.

4.7 Morphological processing

Segmentation results may cause some pixels to be removed producing holes to some parts of the images; this could be seen from characters having some holes in them where some of the pixels were removed during thresholding. Larger holes can cause characters to break into two or more parts/objects. On the other hand, the opposite can also be true, as segmentation can join separate objects making it more difficult to separate characters; these solid objects resemble blobs and are hard to interpret. The solution to these problems is Morphological Filtering. Useful techniques include erosion and dilation, opening and closing, outlining, and thinning and skeletonisation. These techniques work on binary images only. (Phillips, 2000)

4.7.1 Erosion and dilation

Dilation and Erosion are morphological operations which increase or decrease objects in size and can be very useful during the preprocessing stage. Erosion makes an object smaller by removing or eroding away the pixels on its edges; however, dilation makes an object larger by adding pixels around its edges. There are two general techniques for erosion and dilation these are: the threshold and masking techniques. The threshold technique looks at the neighbours of a pixel and changes its state if the number of differing neighbour pixels exceeds a threshold. Basically, if the number of zero pixels in the neighbourhood of a pixel exceeds a threshold parameter then the pixel is set to zero. Fig. 6 shows the result of eroding the rectangle using a threshold value of three (Russ, 2007).

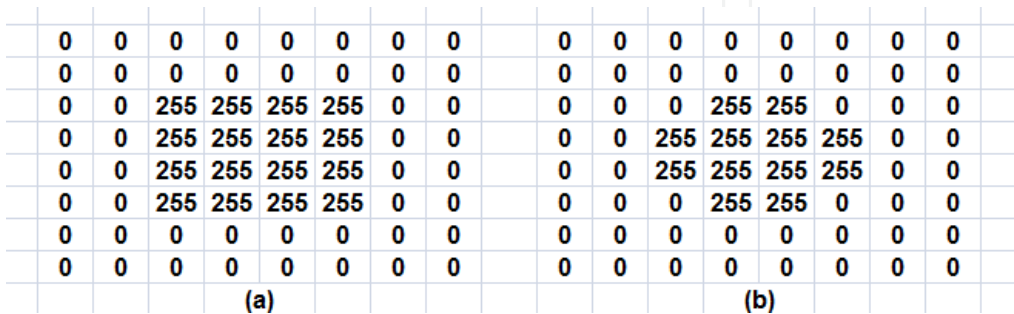


Fig. 6. The result of eroding a rectangle using a threshold of 3.

The dilation process does the opposite of erosion. It counts the value of pixels next to a zero pixel, if the count exceeds the threshold parameter, then the zero pixel is set to the value of the pixel. The dilation in Fig. 7 uses a threshold value of two.

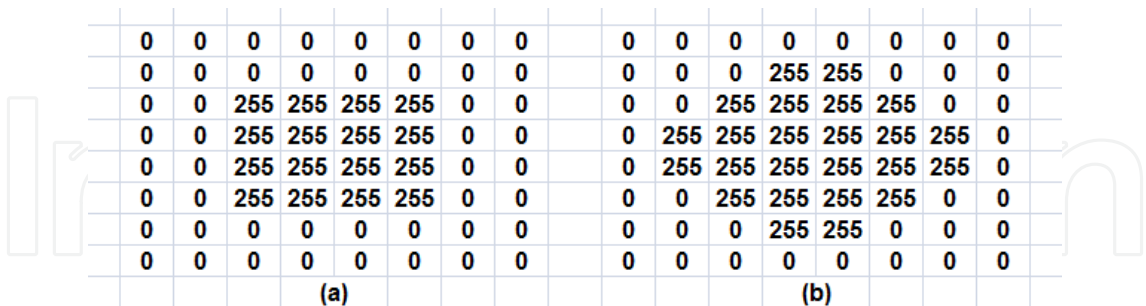


Fig. 7. The result of dilating (a) is given in (b) using a threshold of 2.

The masking technique uses an $n \times n$ (3×3 , 5×5 , etc.) array of 1s and 0s on top of an input image and erodes or dilates the input. Using masks, the direction of erosion or dilation can be controlled. Square masks are more widely used such sizes are 3×3 , 5×5 , 7×7 ... etc. with other sizes could be used (Myler & Weeks, 1993, Phillips, 2000). Masks of sizes 3×3 in different directions are shown below:

vertical mask	horizontal mask	horizontal and vertical masks	
0 1 0	0 0 0	0 1 0	1 1 1
0 1 0	1 1 1	1 1 1	1 1 1
0 1 0	0 0 0	0 1 0	1 1 1

Fig 8. below shows the result of dilation using the horizontal mask.

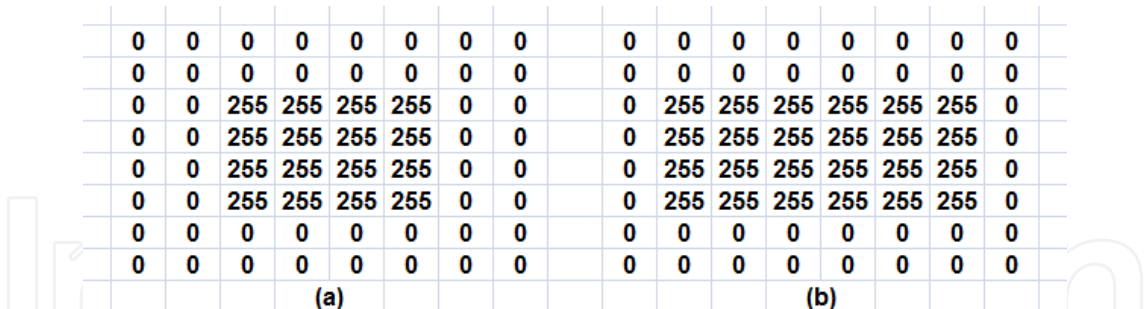


Fig. 8. The result of dilating (a) using the horizontal mask is shown in (b)

Mask erosion is the opposite of dilation. It applies an $n \times n$ mask on the image so that the center of the array is on top of a zero. If any of the 1s coefficients in the mask overlap a white pixel (255) in the image then it is set to zero. Vertical mask erosion removes the top and bottom rows from an object, horizontal mask removes the left and right columns and the horizontal and vertical masks remove pixels from all edges.

To conclude, dilation causes objects to grow in size as it will exchange every pixel value with the maximum value within an $n \times n$ window size around the pixel. The process may be repeated to create larger effects. However, erosion works the same way except that it will cause objects to decrease because each pixel value is exchanged with the minimum value within an $n \times n$ window size around the pixel (Phillips, 2000).

4.7.2 Opening and closing

Opening and closing are morphological operators that are derived from the fundamental operations of erosion and dilation, and are normally applied to binary images. The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground pixels from the edges of regions of foreground pixels. However, it is less destructive than erosion in general. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground regions in an image, but it is less destructive of the original boundary shape.

Opening spaces objects that are too close together, detaches objects that are touching and should not be, and enlarges holes inside objects. Fig. 9 shows two objects joined by a thread; opening was used to remove this thread and separate the two objects, thus, by eroding the object twice the thread is erased. In this case, dilation would enlarge the two objects back to their original size, but does not re-create the thread (Phillips, 2000).

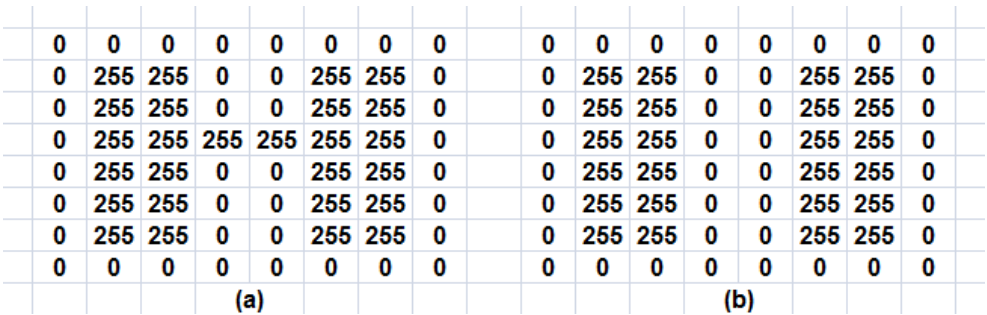


Fig. 9. The result of opening two objects joined by a thread

Opening can also enlarge a desired hole in an object; it involves one or more erosions followed by a dilation process. Closing joins broken objects and fills in unwanted holes in objects, Fig. 10 shows two objects that should be joined to make a line and Fig. 11 shows how closing fills a hole in an object.

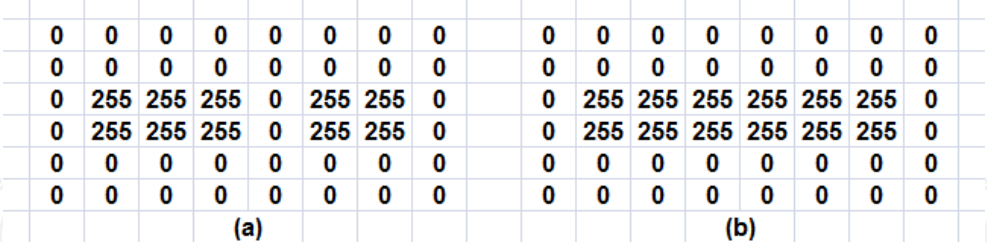


Fig. 10. The result of closing unwanted holes in objects to form a line.

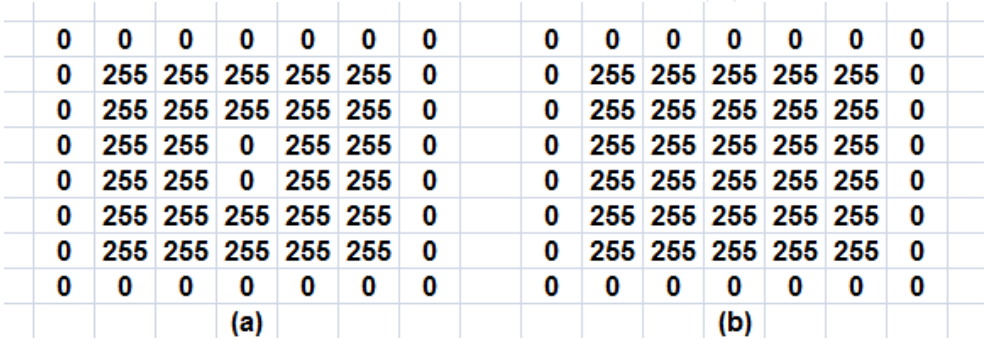


Fig. 11. The result of closing unwanted holes in objects.

The opening and closing operators work well, but sometimes produce undesired results where closing may merge objects which should not be merged and opening may enlarge holes and cause an object to break. The answer is special opening and closing that avoid such problems, for further information the reader is referred to (Phillips, 2000; Russ, 2007; Gonzalez et al., 2004).

4.7.3 Outlining

Outlining is a type of edge detection; it only works for binary images, but produces better results than regular edge detectors. Outlining binary images is quick and easy with erosion and dilation. To outline the interior of an object, erode the object and subtract the eroded image from the original, for example Fig. 12. To outline the exterior of an object, dilate the object and subtract the original image from the dilated image, for example Fig. 13. Exterior outlining is easiest to understand where dilating an object makes it one layer of pixels larger and subtracting the input from this dilated larger object yields the outline.

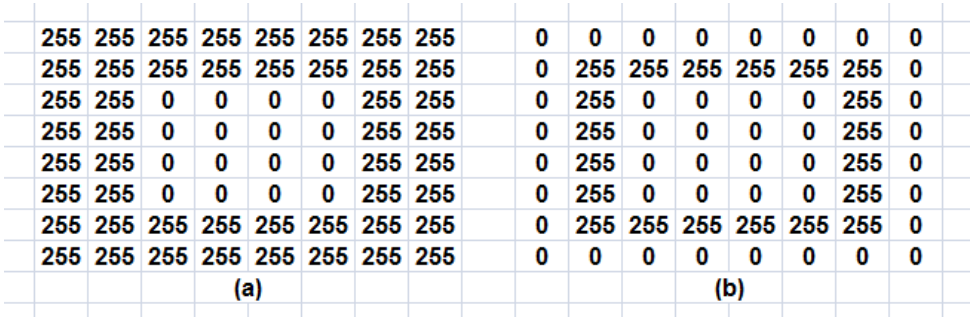


Fig. 12. The result of showing the interior outline of an image.

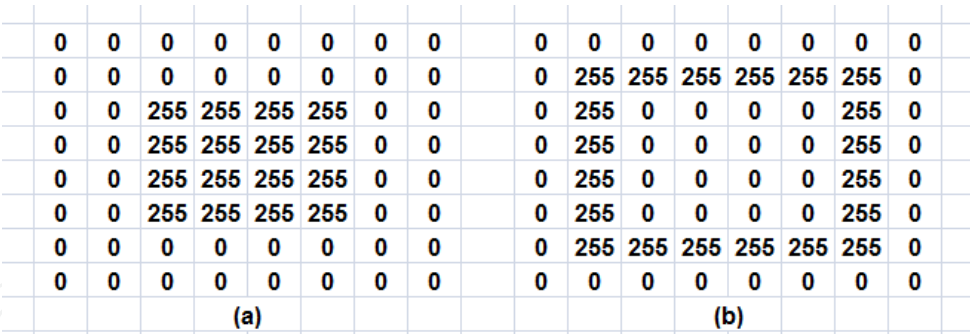


Fig. 13. The result of showing the exterior outline of an image.

4.7.4 Thinning and skeletonisation

Skeletonisation is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while removing most of the original foreground pixels. It is clear to imagine that the skeleton is as the loci of centres of bi-tangent circles that fit entirely within the foreground region being considered, this can be illustrated using the rectangular shape in Fig. 14.

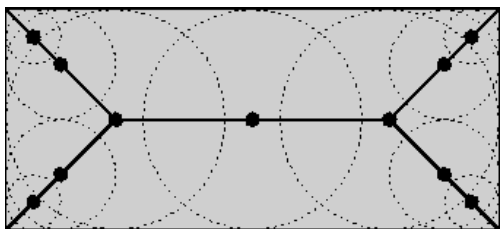


Fig. 14. Illustration of the concept of skeletonisation

There are two basic techniques for producing the skeleton of an object: basic thinning and medial axis transforms. Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. Thinning is a data reduction process that erodes an object until it is one-pixel wide, producing a skeleton of the object making it easier to recognize objects such as characters. Fig. 15 shows how thinning the character E produces the skinny shape of the character. Thinning is normally only applied to binary images, and produces another binary image as output. Thinning erodes an object over and over again (without breaking it) until it is one-pixel wide. On the other hand, the medial axis transform finds the points in an object that form lines down its center (Davies, 2005).

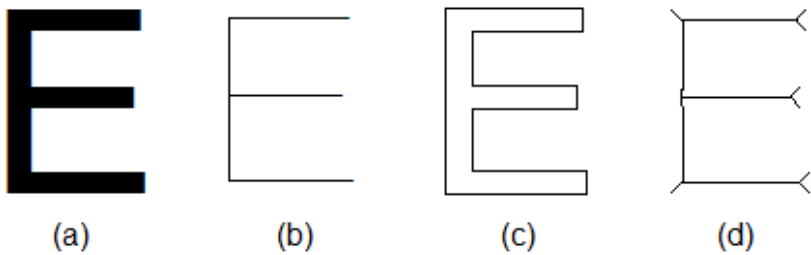


Fig. 15. (a) Original Image (b) Medial Axis Transform (c) Outline (d) Thinning

The medial axis transform is similar to measuring the Euclidean distance of any pixel in an object to the edge of the object, hence, it consists of all points in an object that are minimally distant to more than one edge of the object (Russ, 2007).

5. Conclusion

In this chapter, preprocessing techniques used in document images as an initial step in character recognition systems were presented. Future research aims at new applications such as online character recognition used in mobile devices, extraction of text from video images, extraction of information from security documents and processing of historical documents. The objective of such research is to guarantee the accuracy and security of information extraction in real time applications. Even though many methods and techniques have been developed for preprocessing there are still problems that are not solved completely and more investigations need to be carried out in order to provide solutions. Most of preprocessing techniques are application-specific and not all preprocessing techniques have to be applied to all applications. Each application may require different preprocessing techniques depending on the different factors that may affect the quality of its images, such as those introduced during the image acquisition stage. Image

manipulation/enhancement techniques do not need to be performed on an entire image since not all parts of an image is affected by noise or contrast variations; therefore, enhancement of a portion of the original image maybe more useful in many situations. This is obvious when an image contains different objects which may differ in their brightness or darkness from the other parts of the image; thereby, when portions of an image can be selected, either manually or automatically according to their brightness such processing can be used to bring out local detail. In conclusion preprocessing is considered a crucial stage in most automatic document image analysis systems and without it the success of such systems is not guaranteed.

6. References

- Alginahi, Y.; Sid-Ahmed, M. & Ahmadi, M. (2004). Local Thresholding of Composite Documents Using Multi-Layer Perceptron Neural Network, *Proceedings of the 47th IEEE International Midwest Symposium on Circuits and Systems*, pp. I-209 – I-212, ISBN: 0-7803-8346-X, Hiroshima, Japan, July 2004, IEEE.
- Alginahi, Fekri & Sid-Ahmed. (2005). A Neural Based Page Segmentation System, *Journal of Circuits, Systems and Computers*, Vol. 14, No. 1, pp. 109 – 122.
- Alginahi, Y. M. (2008). Thresholding and Character Recognition in Security Documents with Watermarked Background, *Digital Image Computing: Techniques and Applications*, DICTA 2008, Canberra, Australia, December 1-3.
- Al-Shatnawi, A. & Omar, K. (2009). Skew Detection and Correction Technique for Arabic Document Images Based on Centre of Gravity, *Journal of Computer Science* 5 (5), May 2009, pp. 363-368, ISSN 1549-3636.
- Davies, E. (2005). *Machine Vision – Theory Algorithms Practicalities*, Third Edition, Morgan Kaufmann Publishers, ISBN 13: 978-0-12-206093-9, ISBN-10: 0-12-206093-8, San Francisco, CA, USA.
- Fischer, S., (2000). *Digital Image Processing: Skewing and Thresholding*, Master of Science thesis, University of New South Wales, Sydney, Australia.
- Gonzalez, R.; Woods, R. & Eddins, S. (2004). *Digital Image Processing using MATLAB*, Pearson Education Inc., ISBN 0-13-008519-7, Upper Saddle River, NJ, USA.
- Horn, B.K.P. & R.J. Woodham (1979). Destriping LANDSAT MSS Images using Histogram Modification, *Computer Graphics and Image Processing*, Vol. 10, No. 1, May 1979, pp. 69–83.
- Hull, J. J. & Taylor, S.L. (1998). Document Image Skew Detection Survey and Annotated Bibliography, *Document Analysis Systems II*, Eds., World Scientific, pp. 40-64.
- Kruatrachue, B. & Suthaphan, P. (2001). A Fast and Efficient Method for Document Segmentation for OCR, TENCON, *Proceedings of IEEE region 10 Int. Conf. On Electrical and Electronic Technology*, Vol. 1, pp. 381-383.
- Mao, S., Rosenfeld, A. and Kanungo, T. (2003). Document Structure Analysis Algorithms: A Literature Survey, *Proceedings of SPIE Electronic Imaging*, pp. 197-207.
- Myler, H. R. & Weeks, A. R. (1993). *Computer Imaging Recipes in C*, Prentice Hall Publishing, Englewood Cliffs, New Jersey.
- Nagy, S. & Stoddard, S. (1986). Document Analysis with an Expert System, *Pattern Recognition in Practice II*, pp. 149-155.

- Niblack, W. (1986). *An Introduction to Digital Image Processing*, Prentice Hall, ISBN-10 0134806743, ISBN-13 : 978-0134806747, Englewood Cliffs, NJ. USA, pp. 115-116.
- Nixon, N. & Aguado A. (2008). *Feature Extraction & Image Processing*, second edition, ISBN 978-0-12-372538-7, Elsevier Ltd., London, UK.
- O’Gorman, L.; Sammon, M. & Seul, M. (2008). *Practical Algorithms For Image Analysis*, Cambridge University Press, ISBN 978-0=521-88411-2, New York, NY, USA.
- Otsu, N. (1979). A Threshold Selection Method From Grey-level Histograms, *IEEE Transactions On systems, Man and Cybernetics*, SMC-9, pp. 62-66.
- Phillips, D. (2000). *Image Processing in C*. Electronic Edition 1.0, 1st Edition was published by R & D Publications, ISBN 0-13-104548-2, Lawrence, Kansas, USA.
- Rosin, P. & Ioannidis, E. (2003). Evaluation of Global Image Thresholding for Change Detection, *Pattern Recognition Letters*, Vol. 24, pp. 2345-2356.
- Russ, J. (2007). *The Image Processing Handbook*, Fifth Edition, CRC Press, Boca Raton, FL, USA.
- Sahoo, P.; Soltani, S. & Wong, A. (1988). A Survey of Thresholding Techniques, *Computer Vision Graphics Image Processing*, Vol. 41, pp. 233-260.
- Tang, Y. Y., Lee, S.W & Suen, C. Y. (1996) Automatic Document Processing: A Survey, *Pattern Recognition*, Vol. 29, No. 12, pp. 1931-1952.
- Tanner, S. (2004). Deciding whether Optical Character Recognition is Feasible, King’s Digital Consultancy Services, http://www.odl.ox.ac.uk/papers/OCRFeasibility_final.pdf
- Trier, O. & Jain, A. (1995). Goal-Directed Evaluation of Binarization Methods, *IEEE Trans. On Pattern Recognition and Machine Intelligence*, Vol. 17, No. 12, pp. 1191-1201.
- Wahl, F. Wong, K. & Casey, R. (1982). Block Segmentation and Text Extraction in Mixed Text/Image Documents, *Computer Vision, Graphics and Image Processing*, Vol. 20, pp. 375-390.
- Zhang Z. & Tan, C. (2001). Restoration of Images Scanned from Thick Bound Documents, *Proceedings of the International Conference On Image Processing*, Vol. 1, pp. 1074-1077.

IntechOpen



Character Recognition

Edited by Minoru Mori

ISBN 978-953-307-105-3

Hard cover, 188 pages

Publisher Sciyo

Published online 17, August, 2010

Published in print edition August, 2010

Character recognition is one of the pattern recognition technologies that are most widely used in practical applications. This book presents recent advances that are relevant to character recognition, from technical topics such as image processing, feature extraction or classification, to new applications including human-computer interfaces. The goal of this book is to provide a reference source for academic research and for professionals working in the character recognition field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yasser Alginahi (2010). Preprocessing Techniques in Character Recognition, Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTech, Available from: <http://www.intechopen.com/books/character-recognition/preprocessing-techniques-in-character-recognition>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen