Abstract

Segmenting a scanned document is an essential step in the optical character recognition (OCR) process to recognize the text in a scanned document. The scanned image may have several blocks of text as well as non-text areas. The text regions when surrounded by a rectangle each may not always have non-overlapping rectangles but overlapping rectangles or rectangles within other rectangles. The text regions also may have text of different sizes and fonts which makes it necessary to process each region separately when performing line segmentation and character recognition. In this paper, we propose a method and system for segmenting such images into isolated blocks of text and then processing each block separately as if it were a separate page resulting in line segmentation of each block and segmenting each line into words or phrases. The implementation of the proposed algorithm illustrates its robustness nature. The newspaper page images were processed automatically with no constrains.

keywords: OCR; Document Processing; Segmentation; Connected Components; Arabic language

1 Introduction

A typical OCR system usually has three stages: OCR preprocessing stage for extracting the clear retinal text image in order to improve the accuracy of recognition, OCR engine stage for converting text image to digital text, and OCR post-processing stage for improving the accuracy of the recognize text and extract it in an appropriate document format [1].

Automatic conversion of scanned documents into editable and searchable text requires the use of accurate and robust optical character recognition systems, usually referred to as OCR systems. English OCR systems have reached a high level of accuracy due to many reasons. One of the main reasons is the ability to segment images of English text all the way down to isolated characters when the scanned characters are not touching. Touching scanned characters present a challenge to the OCR systems and reduce their accuracy when the pitch is a variable one. Touching characters are easier to segment with high accuracy in the case of fixed pitch fonts such as Courier.

Arabic scanned text is harder to segment to characters due to the fact that characters in Arabic touch most of the time because Arabic is written similar to script in English. Another major reason is that many Arabic letters have some dots and accents like the lower case “l” and “j” in English. The last reason is that Arabic writing can be written with or without the accents that indicate the vowels, which is called “Tashkeel” in Arabic. These are some signs that are placed above or below the letters to indicate the pronunciation of the letter and the vowel that follows it.

While English l can take two shapes: uppercase and lowercase, many Arabic letters have three or four shapes depending on whether they are placed at the beginning of the word, middle of the word, end of the word, or stand alone.

All this made us conclude that it is much more accurate and proper to segment a scanned Arabic document into images of segmented blocks, and then each block into images of segmented lines of text or images of segmented words or phrases of text but not images of segmented characters before passing the words or phrases in sequence to the OCR engine to recognize them [2].

To obtain the best results from the OCR engine the OCR preprocessing stage must be performed to pass a clear and retinal word or phrase to have a recognize text with high accuracy and lowest recognition time [4].

Because the scanned image may have several paragraphs and titles, they will be segmented into blocks, where each block may have different font size and style. The text blocks must be separated to avoid reading lines of neighboring paragraphs as one line which will cause many errors when the text size and alignment in the two neighboring paragraphs or blocks do not match.

Block segmentation also enables the postprocessor stage to post the recognized text into the block designated spaces in the recreation of the document by inserting the recognized text in the corresponding positions of the blocks in the original image [3].

2 Literature Review

Publishers of newspapers want to convert the printed materials into digital resources in order to preserve and provide future easy access of the content to the readership. There are a good number of algorithms available in the
literature that perform page segmentation. These algorithms trace the basic components within the newspaper page. The components comprise of regions, images, drawing and lines. There are a number of significant challenges facing the segmentation algorithms. Among those challenges are the quality deterioration of the scanned newspaper due to time and the complex layout of the newspaper pages [5]. Liu et al. [6] implemented an algorithm which is based on the components bottom-up to detect horizontal and vertical lines. The algorithm is iteratively merging neighboring components such as drawing, photos, titles and inverse titles. Another algorithm is locating rectangular regions within the newspaper page which contain mostly foreground pixels [7]. It is then forming patterns from the located regions and finally performing the classification based on the size of the pattern. Hadjar et al. [8] only detect horizontal and vertical line segments and do not distinguish photos from drawings. They end by merging the neighboring regions.

Antonacopoulos et al. [9] binarized the document using a multiscale implementation, identified and removed vertical and horizontal separators and denoise the document. They analyzed text indentations, spaces between adjacent lines and text line features in order to split regions into paragraphs which are merged together if overlapped significantly. Chen et al. [10] segmented the page into text and non-text blocks. They first binarized the document image, extracted the foreground connected components and formed horizontal and vertical cut rectangles. Finally, the vertically adjacent text lines are linked into text blocks.

Line extraction from text blocks within newspaper pages is an important task towards recognizing the text in the newspaper pages. Gatos et al. [11] assigned values for every pixel based on the horizontal and vertical run-length. Their method tolerated slight line skewing and broken lines.

3 System Overview

The presented OCR preprocessing component is composed of many steps like:

- Converting the color image to a grayscale image.
- Noise removal using grayscale filters.
- Converting the grayscale image to a binary image.
- Noise removal using binary filters.
- Reverse video detection of foreground and background.
- Skew detection from the binary image using a modified Hough transform.
- Calculation of connected components.
- Text and non-text separation.
- Skew detection and correction using the connected components.
- Block segmentation into blocks or paragraphs.
- Line segmentation of blocks.
- Word or phrase segmentations of the line images.
- Some of the text blocks fall within bigger blocks like a text rectangle or circle fall within a larger rectangle of text. The rectangular blocks around the text blocks may overlap making the page more complicated.

The output of this system is a sequence of sorted blocks and a list of line images within each block right side up, non-skewed, noise free, to be passed to the line recognition engine of choice in binary or grayscale formats. The system segments the page into clean segmented text line images which are oriented properly and clean of noise.

It is to be noted that the presented method here assumes no prior information of the page structure or content or the font or the size of the font.

The statistics are performed on the connected components using their parameters such as their dimensions, aspect ratio, density, and perimeter. Then rules are added to filter out obvious non-text components. Statistics are performed on the remaining text text-like components to find the common height of words or lines of the dominant text size even though the page may have other sizes.

Statistics are then run again on components belonging to each single block to use the new block specific word common height for further text and non-text separation in case different blocks have different sizes of text. The word spacing and block spacing are dynamically calculated using this common height. This way, there are no hard coded values and the page calculations are dynamic and do not change if the page has been enlarged or reduced in size. This is achieved by using dimensionless parameters that change with size.

A distance value that is greater than the calculated word spacing is then calculated. It and must be less than the block spacing. Then words are merged if spaced by less than this calculated distance. The word spacing is not used to do word segmentation. A distance is calculated based on the word height and block spacing to be used to segregate the components into blocks. Since some of the blocks may fall within other blocks like a circle text area in the middle of a large blocks or an L-shaped block in any direction with text blocks underneath or above the horizontal part. This could make the rectangles surrounding each block overlap.

The separation of components into blocks that could overlap is done by merging a component with a block if the closest text component to this component falls within that block and the distance is smaller than the margin calculated above. If we merge the cell with the closest block all overlapping blocks would become one major block. So, merging is done to the block that hosts the closest component to the current component.

Post-processing is used to merge blocks that should be merged into one. For instance, blocks that are layers above one another with similar font sizes and approximate dimensions are be merged into one block. The blocks are sorted based on their location on the page (their
coordinates) to help stream the recognized resulting text in the right order. The algorithm goes top down and left to right making use of some wide blocks that may span the entire width or height of the page or most of it, etc. Also it makes use of non-text lines that may exist and separate blocks.

Figure 1: Image containing multiple blocks of text

Post-processing for each block is also used to check for touching lines within the block and then separates them. Statistics are calculated to find the average line height. If the line thickness is larger than the average line height by multiple times, the line is split into several lines at the horizontal places of least overlapping between the consecutive lines.

2.1 Multiple Block Images

For complex scanned images with multiple blocks of text, the blocks when represented by rectangles will be overlapping. The task here is to segment the image into blocks as in Figure 1 then if some blocks belong to one block they are merged which will result in a segmented image with less number of blocks as shown in Figure 2.

2.2 Block Segmentation Operations

The system performs some sequential operations on the input image to finally produce the clean and properly oriented segmented line images. These sequential steps include image binarization, reverse video detection and correction, noise removal, skew detection and correction, connected components calculations, line height calculations, word spacing calculations, line spacing calculations, removal of frames and straight lines, word merging and non-text merging, separation of text and non-text, recalculation of line height, word spacing, and line spacing, line segmentation, and touching line separation:

• Initial Line Height Calculations: Same as in the single block case as if the page has the same size in all blocks.
• Minimum Block Spacing Calculations: Experimental results of pages with multi-blocks and different sizes of fonts have been calculated. Then, modeling of the minimum block spacing is calculated as a factor of the dominant line height calculated above. This value is measure for many pages and modeled, then the dynamically calculated ones are compared for the measured ones for calibration of the value.
• Removal of Large Frames and Straight Lines: It is necessary in this step to remove large frames and straight lines that not within a single block; i.e. they may surround many blocks or separate blocks from one another. Special rules were designed for removing such frames and lines using the dynamic initial line height calculated above.
• Connected Component Merging into Blocks: All the connected components that are horizontally or vertically spaced by less than the minimum blocks spacing calculated above are merged into one group, which is a block. The distance is measure to the nearest neighboring component in each block and not
to the bounding rectangular box of each block. This will result in one or more blocks. Some blocks may end up being split into many if the paragraphs are separated by a large spacing.

- Block Merging: A post processing step is performed to merge blocks that seem to belong to one block. They are usually horizontally overlapping rectangles with comparable average connected component size and close to one another with no separation in between them.

- Block Sorting: The blocks are sorted based on their location on the page (their coordinates) to help stream the recognized resulting text in the right order. The algorithm goes top down and left to right making use of some wide blocks that may span the entire width or height of the page or most of it, etc. Also it makes use of non-text lines that may exist and separate blocks.

4 Block Segmentation Method

The corpus built to assess the algorithm performance includes a collection of more than 100 pages from 10 Arabic newspapers beside King Abdulaziz City for Science and Technology (KACST) newsletter. It is a large-scale, lexicon-free, omnifont and multi-style line-based corpus. The newspaper pages were then scanned and saved as separate documents. Each page image was segmented into a sequence of text and non-text blocks. Text blocks were further segmented into paragraphs based on the alignment of the newspaper. Around 600 paragraphs were extracted from the newspaper page images where each text block might include one or more than one article. Those paragraphs were segmented into more than 26000 text line images with image heights varying between 8 and 173 pixels.

Segmenting images like that shown in Figure 5, with blocks that could fall within one another or curve around one another is more complicated than images with blocks that can be represented by non-overlapping rectangles.

Experimental results of pages with multi-columns and different sizes of fonts have been calculated. Then, modeling of the minimum column spacing is calculated as a factor of the dominant line height calculated above. This value is measure for many pages and modeled, then the dynamically calculated ones are compared for the measured ones for calibration of the value. A real examples of a complex page is shown in Figures 3, 4, 5, and 6.

5 Conclusions

In this paper, we proposed a method to segment a complex scanned Arabic document that has multiple overlapping blocks of text in different shapes and fonts and font sizes. The output of the system is a list of sorted blocks and for each block a list of line images and or each line a list of word and phrase images per line.

Acknowledgment

This research is fully funded by King Abdulaziz City for Science and Technology (KACST).

References

Figure 3: Original scanned image after converting it from color to grayscale

Figure 4: The grayscale image after skew detection and correction


Figure 5: The image after block segmentation

Figure 6: The image after each block is segmented into lines and phrases