Islamic University of Gaza
Faculty of Information Technology
Deanery of Postgraduate Studies

Arabic Typed Text Recognition in Graphics Images (ATTR-GI)

This thesis Submitted to the Faculty of Information Technology. Islamic University of Gaza

In Partial Fulfillment of the Requirements for the Degree of Master of Information Technology

Prepared By:

Lamiya Mohmmed El_Saedi

Supervisor

Dr. Ashraf Alattar

September

2013

بسم الله الرحمن الرحيم

﴿قُلْ لَوْ كَانَ الْبَحْرُ مِدَادًا لِكَلِمَاتِ رَبِّي
لَنَفِدَ الْبَحْرُ قَبْلَ أَنْ تَنْفَدَ كَلِمَاتُ رَبِّي وَلَوْ
جِئْنَا بِمِثْلِهِ مَدَدًا﴾

[الكهف١٠٩]

Dedication

I dedicate this research

To the spirit of my dear mother Asma'a Syam

To my dear father Mohmmed El-Saedi,

To all my family and all my friends

To everybody who prayed for me

To my university

To my our colleagues

I dedicate this research

Acknowledgment

First of all I would like to thank Allah for help me and get me a long patient to complete this research.

Secondly I would like to thank my supervisor Ashraf Al_Attar because he is very challenger to exit this research in this picture.

I would like to thank my husband Ashraf Salama and my child because they suffered with me so much.

I would like to thank every people whom supported to me and particularly Dr.Ala'a El_Halees, Dr. Eyad El_Agha, Dr. Rawia Radi, Miss. Marwa Abo Jalala, and Miss. Sarah Kuhail.

I want to thank my father and my sisters Sana'a and Shyma'a, also my brother Abdullah for supported me for all time.

I don't forget to thank Islamic University and all teachers in faculty of Information Technology specially dean Dr. Tawfieq Barhoom.

Finally I would like to thank all my friends and everyone helps this research to see the sun light.

Abstract

While optical character recognition (OCR) techniques may perform well on standard text documents, their performance degrades significantly in graphics images. In standard scanned text documents OCR techniques enjoy a number of convenient assumptions such as clear backgrounds, standard fonts, predefined line orientation, page size, the start point of written. These assumptions are not true in graphics documents such as Arabic advertisements, personal cards, screenshot. Therefore, in such types of images, greater attention is required in the initial stage of detecting Arabic text regions in order for subsequent character recognition steps to be successful. Special features of Arabic alphabet characters introduce additional challenges which are not present in Latin alphabet characters.

In this research we propose a new technique for automatically detecting text in graphics documents, and preparing them for OCR processing. Our detection approach is based on some mathematical measurements to know is it a text or not and to know is it Arabic Based Text or Latin Based. These measurements are follows, measure the Base Line (the line has maximum number of black pixels). Also, measure Item Area (the content of extracted sub images). Finally, find maximum peak for the adjacent black pixels in Base line and maximum length for sub adjacent black pixels. Our experiment results will come in more details.

We believe our technique will enable OCR systems to overcome their major shortcoming when dealing with text in graphics images. This will further enable a variety of OCR-based applications to extend their operation to graphics documents such as SPAM detection from image, reading advertisement for blind people, search and index document which contain image, enhancing for printer property (black white or color printer) and enhancing OCR.

Keywords: *Arabic Text Recognition ATR, Arabic text information recognition ATIR, Arabic Typed Text Recognition in Graphics Image ATTR-GI, Arabic optical*

character recognition AOCR, Text Extraction, Optical Character Recognition, Text recognition.

Glossary

In our research there are some words we use it in this meaning like:

	Abbreviated	Meaning
1	**Object**	This word we use it to refer for entire image.
2	**Item**	This word we use it to refer the elements in entire image.
3	**Region**	Is a rectangle area that contains one statement or word or character

Table of Contents

List of Figures:

List of Tables:

No table of figures entries found.

Chapter 1

Introduction

This chapter contains nine sections. We will declare some definitions for abbreviations or terms that are related to our work, starting from OCR definition until we even got a way to draw the appropriate definition of AOTR. Then we display problem statement, objective, scope and limitation, significance of thesis, and research format respectively.

1.1 Optical Character Recognition

Optical Character Recognition abbreviated (OCR), defined in [44] and [49] as the process to convert the scanning image of papers to new documents can be easily used with computers for manipulations. This process can dealing with handwritten or printed characters. Also, it is useful for commercial and education to simplify for getting information like credit card in the Banks or shops. OCR used in two ways according the components of the system, i.e. some of them needs Hardware and Software, and the other needs only Software system.

Another definition defined by AIM in [3] as "OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism."

Also, it defined by Abuhaiba in [1] as "is the process of converting a raster image representation of a document into a format that a computer can process. It involves many subscriptions of computer science including image processing, artificial intelligence and data base systems."

So, we can define OCR as a technique used as part of recent series of operations to recognized handwritten or printed (typed) characters. This needs learn the system how to recognize different characters. And possible to make it part of machine learning.

Optical character recognition (OCR) has been a topic of interest since possibly the late 1940's when Jacob Rabinow started his work in the field [38]. The OCR research started in 1960's and addressed at first for reading Latin characters. After ten years the first paper on Arabic word recognition was published [32]. The OCR system can broadly categorize into two categories: on-line and off-line OCR systems [24]. The on-line OCR defined by [24] as "the recognition is performed at the time of writing as it is the case in PADs, needs some information like starting and ending points of the character, and usually facilitate writing separate characters, not complete words. But off-line OCR deals with scanned images."

1.2 Optical Text Recognition

Optical Text Recognition or Text Recognition abbreviated (OTR/TR) and OCR both of them are an active area in both academic research and commercial software development [34]. Farther more, shared with general steps for recognition process like (1) preprocessing: prepare the image to be a binary image not colored image to make the work on image is easier, (2) text extraction: that's mean get the pixels that performs character text to be an independent part from the rest of the image, (3) segmentation: if we have a long text with white spaces, then segment each word or sub-word as a single image, (4) separation: in a single word that does not have a white space, make each character as an individual character in a single image, and (5) character recognition: identify the character by comparing it with a set of character for recognizing process. But in OTR focus on the steps before character recognition which that's mean the steps from (1) to (3). In other words, OTR based on recognition of text at first by significant the text region with rectangle shape. If the development completes the process with steps (4) and (5) then they are going to recognize separate characters and this is OCR.

Most researches in OCR or OTR have been on Latin base characters, with some researches on Arabic, Chinese and Japanese character recognition [9].

Effective text recognition techniques are widely used, such as for indexing and retrieval of document images and understanding of text in pictorial images or videos [34]. In other words, text recognition allows translating images of text as scanned documents or images of natural scenes containing signs into actual text characters. This is only true for clean, non-distorted images [18].

From above definitions there is no specific topic called text recognition as itself. Searching on text recognition is almost take results on optical character recognition with little text recognition. That means the two titles are the same meaning and contains the same methodology to recognize finally characters in text image (image that already contains text). The text recognition step is used as text localization to know the first location of the text only assuming the text exists in the image. Then complete the rest of recognition which are separation, segmentation then finally character recognition or word recognition.

Most researches in text recognition or OCR depends on comparing between set of words / characters respectively to identify the word or character in the scanned paper or graph image but not necessary to know if this region is a text before recognized it. So, it takes a lot of times for comparing process and if the word is not found in the data set, the model does not recognize the word. So, for this reason they need a huge data to make the search process more accurate.   Michael et. al. [18] worked on English language are depend on search process by connected to Google and Yahoo search engine on two levels. The first one is word level, and the other is statement level. Because these search engine contains a huge number of words to improve the recognition process. The meaning of Text Recognition in [18] passes through the comparison process between stored words that used for recognition and they decided if this part of image is a text or not. So, text recognition going through the following phases: (1) Text Localization, (2) Segmentation, (3) Character Recognition, and (4) Contextual post processing. The researcher should go through these all steps when the researcher needs to make text recognition.

But in our work we need to solve a big challenge which that how to recognize the set of pixels grouped as a region if it is a text or not (i.e. only focus on preprocessing, text extraction, post processing to be ready to use in other application) by search for a good features and test it to perform a rule that suit for any Arabic Text. To make Text recognition phase as an individual phase, it must be done to comfort the researcher before that to complete next phases.

In other words we can say, we try to separate text recognition step from the rest of steps. By making text recognition as a special stage, separate and essential to accomplish the rest of the tasks. This will cause a paradigm shift so that makes it easier for researchers to focus only on the part of the solution. So that, development can be easily and become an operations molecules can be linked with any appropriate portion, which increases the efficiency and improvement in test results the appropriate section and link it with the rest of the parts.

So that, we will present more definitions and details that are supports and describe our idea.

In next two sub sections we try to clarify the meaning of Text Information Extraction (TIE) and Arabic Optical Text Recognition (AOTR). Also present some challenges in this area.

1.3 Text Information Extraction, and its stages

Kim said in [11] "Text extraction research can be divided into two groups: graphic text [7], [37] and scene text [12], [23] and [38] extractions. Jain *et al*. [7] extracted graphic texts in various images: binary, web, color and video frames. Gray-level values and color continuity were used as features. The method had good performance for binary, web, and video images, but it had poor results on color images."

To extract text from an image you need as a first step (zoning) that defined in [34] as "which analyzes the layout of an input image for location and ordering the text blocks. Then each text blocks containing homogeneous text lines of the same orientation is processed for text recognition. However, this zoning approach cannot handle documents that don't have homogeneous text line, such as artistic documents, pictorial image with text, raster maps and engineering drawings."

In [10] Keechul et. al. defined Text Information Extraction (TIE) as a process might be used in set of scenes or in fixed image. It goes through the following phases: (1) Text detection, to determine the presence of text in sequence of images. (2) Text localization, to determine the location of text in the image and generating bounding boxes around the text. (3) Text tracking, used to reduce the processing time for text localization and to maintain the integrity of position across adjacent frames. (4) Text extraction is the stage where the text components are segmented from the background. (5) Enhancement, the text separated from image required an enhancement because the text region usually has low-resolution and is prone to noise. After that, the text was extracted ready for using in OCR technology, as shown in (figure 1.1).

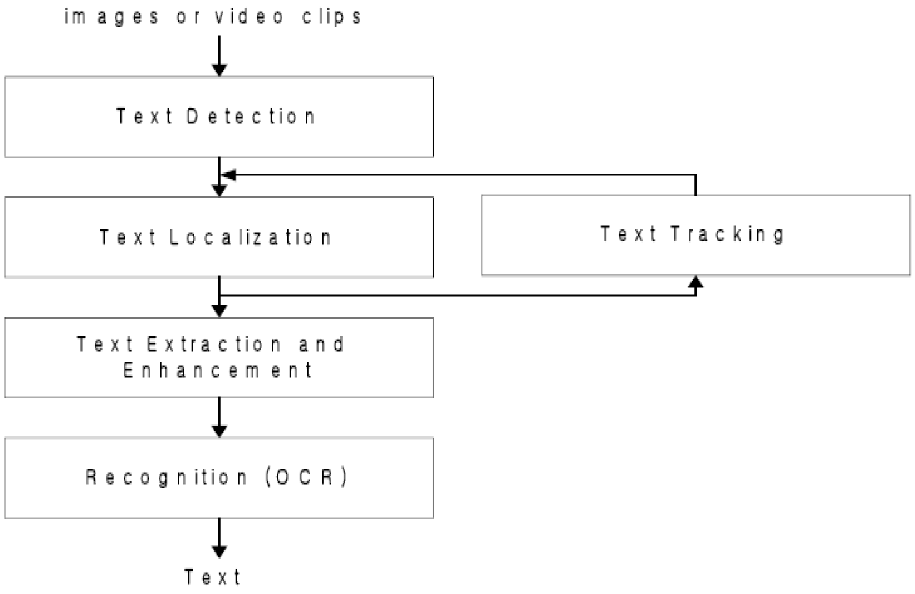


Figure 1. 1 : the architecture of TIE system

In chapter related works we show that there is a significant amount of research works in the literature which has handled this problem to a great extent but mainly for Latin text, but no equally adequate treatment of Arabic text can be found in the literature. Arabic text poses an additional set of different challenges to the problem. Obviously, whether Arabic or any language, the challenge is further complicated when the text is handwritten.

In our research we focus on Arabic typed text in still images, and limit the processing on the first three stages of the TIE process; namely: detection, localization, and extraction. We may extend our work in the future to handle the problem in image sequences. At this extent presented thus far the problem can be referred to as Arabic Optical Text Recognition (AOTR), and we discuss it in more detail in the remainder of this introduction.

1.4 Arabic Optical Text Recognition and challenges

AOTR; Arabic Optical Text Recognition is a branch of OCR specified in Arabic language. It is include on-line and off-line reading technology for handwritten text. That used normal paper or electronic media. The application of optical character recognition was using this technique in many familiar live areas like, check sorting in banks, zip code reading, mail sorting, providing assistance to blind people, reading of customer filled form, automatic office archiving and retrieving text, and improving human-computer interfaces like pen based computers.[9]

The first published work on AOTR is in 1975 by Nazif in [22] as a master's thesis. Developed system for recognizing printed Arabic character based on extracting strokes, that he called radicals, and their position. This kind of research faces many challenges. The challenges in AOTR are related on:

- The different shapes of Arabic characters,
- The overlapping between characters when it's written as a printed computer character or handwritten character,
- Diacritics and variety of Arabic font. [9]

We make a comparison between the standard or traditional image among Scanned pages, that used to insert paper into computer to use it as a data set to approve his/her technique which is the only way to repair degraded papers. Papers with A4 size are easy to know the start point, the spaces between lines. It can be easily identify the text from other part of the scanned paper because it has a white background and a black foreground. Also, the text’s size, type orientation is known.

The other type which is newest one is a graphics image. The image is ready in a computer with variant background color, also includes different objects with different shapes. With or without text. If it has a text, not necessary to start every time at the same point. In addition could be written horizontally, vertically or in diagonal. Other problems are the size, color and type of text. All of these things are obstacles in the identification of the text and added to above challenges, as shown/depicted in (table 1.1).

Table 1. 1: A comparison between Scand pages and graphics advertisement

		Scand pages	Graphics ads
Background color:	**1. Color**	*White*	*Any*
	2. Noise (other elements in background)	*unlikely*	*Very likely*
Lines:	**1. Exist**	*Yes*	*Not necessary*
	2. Start	*Known*	*Not known*
	3. Orientation	*Horizontal*	*Any*
Fonts restrictions:	**1. Type**	*Yes*	*No*
	2. Size	*Yes*	*No*
	3. Color	*Yes (mainly black)*	*No*

1.5 Problem Statement

The problem of this research is how to increase the performance and accuracy of identifying Arabic text from graphics image. The process must be done automatic without select the text in graphics image.

1.6 Objectives

Our objective is divided to main and specific objectives. Specific objectives are much closed to main objective and give more details for our main objective.

1.6.1 Main Objective:

The main goal of this research is to develop a method for filtering, detecting and extracting Arabic text from graphics images. The output of the method is a white background image that contains black Arabic text.

1.6.2 Specific Objectives:

- Collecting suitable data that contains Arabic and Latin based text with colored background and different shapes to improve our idea and to solve the problem of text extraction from complex image.
- Select suitable features to be available for maintenance.
- Develop the method to Extract Arabic text from graphics image.
- Evaluate our module by calculate the rate of text that recognize correctly.
- Integrate our method in an open source Text Information Extraction system

1.7 Scope and Limitation:

- Cover images with both clear as well as cluttered backgrounds.
- Covers horizontal text orientation only.
- Will not include character recognition; only post process in order for OCR algorithms to handle this part.
- Our module is off-line system.

- Will not include handwritten characters.
- Will not include English.
- Will not include region that contain more than one statement.
- Integration will be limited to either OCR system.

1.8 Significance of the thesis

- The set of features extracted about the text region can be utilized to reconstruct the text for editing purposes.
- The algorithm could be used as additional properties to the printer as helper to decrease the amount of ink via remove the pictures and print only text, also make the font size thinner than in the origin image.
- The most recent application of Arabic OCR discussed a one problem that is to rescanning the old/damaged papers to extract text and save it in new documents.
- Prove that there are different ways to use OCR techniques for enhancement like extend OCR into graphics.
- The proposed research can be used in Spam Detection from image. In the past, the ads are sent as a text/document file but now sent as images via e-mail. Companies used this technique to send thousands of messages as a shape of advertisements. This way maybe closed the personal mail box, as a reason for a lot of messages received in mail box and calls it a Daniel of Service which makes a problem for users that prevent him to receive or send any more or important mail messages. In fact there is a SPAM filter to prevent these messages. One of the most common ways to prevent Spam is to use data mining method to classify the message contain as Spam or not Spam. To overcome these techniques, spammer sends the Spam messages as images. SPAM filter can't detect the SPAM message that send as an image message which is a new way used by companies to send it's advertisements through electronic mail. So, SPAM filter may prevent all SPAM image messages depending on the conditions and rules that use it to identify if the message is spam or not. SPAM image message is an image contains background color, graphic shapes,

photos and Arabic/English text with various formats. This kind of image used in advertisement and call it Poster or Business card. When we need to use it in an Internet must be have 72dpi as a resolution, about 400-600 pixels wide for large image; 100-200 for thumbnail image. Preferred file format is JPEG, and the approximate file size 20-200 KB. [33]

- Enable people with limited vision to read Arabic text from image.
- A reusable component that can be used with other applications or devices which depends on extracting textual content from images, and we can call it Friendly-OCR.
- Enhance search engine capabilities in dealing with images:
    - Find images which contain specific words.
    - Find images which contain alias words.
    - Index document which contain images.
- Make dealing with softcopy easier, such as reading, copy, cut, delete and print.

- OCR is already being used widely in the legal profession, where searches that once required hours or days can now be accomplished in a few seconds. [49]

1.9 Research Format

The research is organized as follows. Chapter one is introduction, research problem, objectives, scope and significant. Chapter two is related work. Chapter three is technique and implementation. Chapter four is evaluation and discussion. Chapter five presents conclusions and future work.

Chapter 2

Background and State of the Arts

In this section we review a number of research works in OCR/OTR. We start with non-Arabic works in order to cover the techniques used to deal with the general challenges, and then move on to some Arabic research works to cover techniques which handle the specific problems related to Arabic text.

2.1 Non-Arabic OCR/OTR research work

Kim *et. al.* [11] combined two different levels of text features. These two levels are Low-level and High-level. Low-level contains three image features that used, (1) to find local variation of intensity, (2) colors, and (3) to find color continuity of the same text area. High-level is used to verify the candidate text region by examining stroke composition. Finally, verify character recognition as shown in (figure 2.1). Also, used SVM that works on variant size, for verification by input all previous features to SVM to classify the local area is text or non-text. Their method is just for extracts and verifies longest text regions before the final text recognition with OCR. They found that the average of results be increased in the colored image to become 88.6%. But text image become 85.5%, also test their model on Camera but did not work efficiently. So, they need to improve their model to work in a proper way with Camera. But not experiment on Arabic Text. This emphasizes the process of integration might be done, and this technique suitable for special purpose and might not give a good results in scanning images.

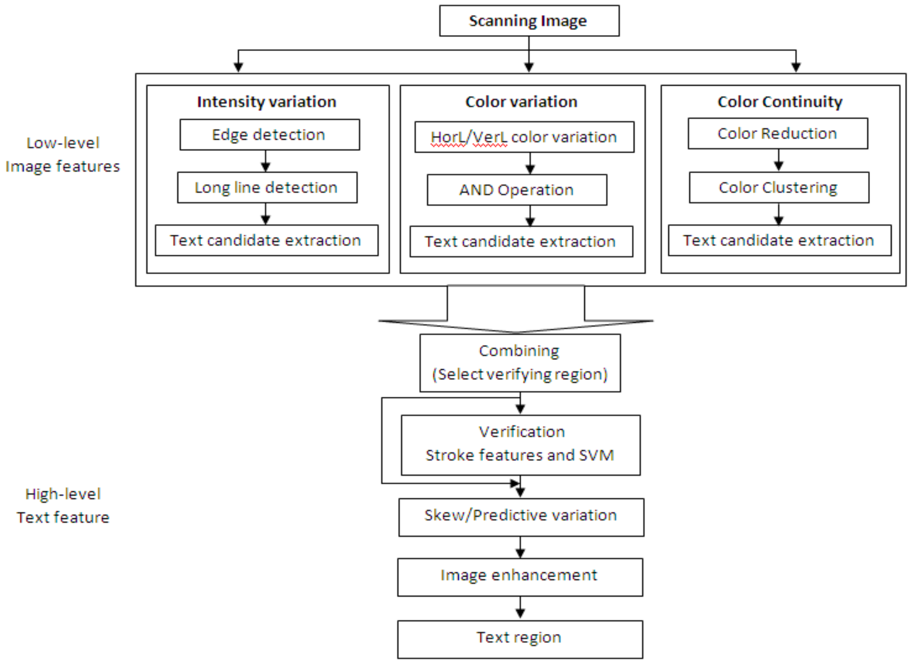
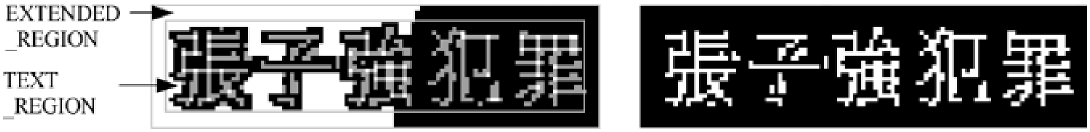


Figure 2. 1: the proposed method for Kim et. al. in [11]

Michael *et. al.* in [18] develop a framework based on identifying the text place automatically in the image by using Maximally Stable Extremal Regions (MSER) detection [16]. This is the best way that interest in Region identification in computer vision according evaluation in [20]. This method depends on Continuous geometric transformations and is invariant to affine intensity changes. Farther more identify the text in different size. The new approach for them is depending on increase the accuracy for word recognition with 87.05% instead of 68.58% which based on ICDAR 2005. So, they connected with web search engine like Google and Yahoo. This way is automatic but need a large number of words to give a correct recognition. But in our thesis we based on the external features to recognize the text.

Michael *et. al.* in [19] performs a detailed analysis of multilingual text characteristics, including English and Chinese. They work on a comprehensive, efficient video text detection, localization, and extraction method. Text detection based on edge detection, local thresholding, and hysteresis edge recovery. Text localization performed to identify text region accurately as depicted in (figure

2.2). Text extraction consists of adaptive thresholding, dam point labeling, and inward filling.



(a) Result of labeling dam points (in gray)

(b) Result of inward filling

Figure 2. 2: represent localize text region and text extraction

This work is much related to our work according the solution technique steps but not in how to implement it. Also, they work on English and Chinese but our work is on Arabic Based only. Other difference is they work on video but our work is on static image.

Yao-Yi *et. al.* in [34] focus the light on the public problem on English text recognition where previous researchers were not going to solve it and tolerated to find a solution for it which that is Multi-oriented, multi-sized, and Curved text. They discussed the problem of text that not written in horizontal but written in vertical or on curved line, and there techniques does not require training for specific fonts and can be easily integrated with a commercial OCR product. They are going to measure the curved angle then put a box on each individual character to make the recognize process easy for him. These are major steps for them approach for text recognition: (1) Extracts the text pixels from the input document, then they have a binary image where each connected component in the foreground is a single character or a part of character such as the dot of the "i". They present the conditional dilation algorithm that depends on Character Connectivity Condition which is "An expansion pixel needs to connect to at least one and at most two characters. This is because the maximum neighboring characters that any character in a text string can have is two.", and Character Size Condition which is "If an expansion pixel connects to two characters, the sizes of the two characters must be similar". (2) Dynamically group the extracted text pixels into the text strings which are the major contribution in his paper. (3) Detect the orientation of each string and rotate it to horizontal direction for text recognition using commercial OCR product like

ABBYY 10 and Strabo systems, but experiments on Arabic Text are not considered.

Zhidong *et. al*. [35] this paper represented a technique to extract a character text from printed document after scanned the document. Used optical character recognition OCR system and hidden Markov modeling HMV technology to model each character, they used this technique on English, Arabic, Chinese and in Fax data. The challenge in this paper is combining between more than one language in HMV model. Because most previous researches used single language in HMV model. The basics system: OCR system includes two parts training and recognition system. The recognition system used the same preprocessing and feature extraction used in training. For feature extraction used feature vectors for each frame (each word or sub-word) and the time is an independent variable. The feature vectors are intensity, vertical derivative for intensity, horizontal derivative for intensity and local slope and correlation across a window of 2-cell square. Combined with the different knowledge estimated from training, were used to find character sequence. They conclude that, there are systems could be work on different language by training on new set of data, based on the HMM model in the OCR system. Also, with training the system could achieve a robust performance on degraded data. The different the system applied on Arabic, English, Chinese and Fax text but we focus only in Arabic. The steps to solve the problem are semi-equal to our steps.

We can say from the above related work, there are two types or levels to verify recognition process. One of them based on text recognition and the other one based on character recognition. Also, they used different features which might be used in our work especially the features mentioned in [11] and [34].

2.2 Arabic Optical character Recognition/Optical Text Recogntion research work

Kasmiran *et. al.* in [9] said that Arabic Optical Text Recognition (AOTR) is a branch of OCR. The first published work on AOTR is back in 1975 as a Master's thesis that developed a system for recognizing printed Arabic characters based on extracting strokes and their position by [22]. In last three decades most of researches were focused on how to enhance the module of Arabic Character Recognition from three sides (1) accuracy, (2) performance, and (3) decreasing the time of character detection. Start with handwritten detection and finish with reading text file to rewrite make a copy …etc. Therefore, all enhancement techniques have a similarity in a base strategy which summarized as follows: (1) Pre-processing, (2) Segmentation, (3) Feature extraction, (4) Classification and (5) Post-processing. Step 4 and 5 were applied according the kind of technique. The Arabic OCR for printed text was a research topic started in the 1990s [29].

In the next few lines we illustrate some researcher's work used the above steps to recognize Arabic typed characters in A4 scanned image file with white background and black foreground. Furthermore, they didn't need to detect if it has a text or not because they sure it contains text and start the seek process from right to left. But they face some problems to define Arabic features extraction likes, Base-line is defined in [9] as "the line on which all letters lie", which use it to determine the space between lines and the distance between characters in the word. In addition, the orientation of text is needed to be specified.

Abuhaiba in [1] addressed the problem of line segmentation and character separation for discrete Arabic script documents. This study based on scanned document as a dataset to experiment his algorithm. The success rate of applying the new algorithm is 94.4%, and the average time required to segment one character was 62 msec. In his experiments Simplified Arabic and Traditional Arabic used as two basis fonts by developing two discrete versions discrete Simplified Arabic and discrete Traditional Arabic. There is no similarity between our approach which is text recognition and the Abuhaiba's approach is character segmentation. Also, the differences is that we don't need to scan the paper to insert into a computer, and repair the resolution of the text to give best result because the image is ready and stored by default in a computer and has a good resolution. Another thing is in our work there is no standard font-style or font-size and the background contains a various noise.

Ahmed *et. al.* in [2] represent a framework to segment Arabic Text, also to extract features font that used in recognition process. Their work based on the combination between Line Adjacency Graph (LAG) and Base-line features. They test their special algorithm on two sets of text files. The first one includes 31 pages but does not contain diacritics while the second set includes 15 pages which contain diacritics signs. All of these pages were insert into computer through scanning process with 300 dpi. The average of correct script classification rates were 95.2% and 94.1% respectively. The average correct recognition rates were 94.8% and 88.9% respectively. Another experiment was performed on pages were printed by laser printer with 300 dpi. The average of correct script classification rate was 94.6%, and the average correct recognition rate was 93.4%, but they graphic images are not included in their experiments.

Anthony *et. al*. in [5] proposed a serious problem on Arabic recognition that is segmentation. This problem comes from the Arabic text written with overlapping rather than the other language which come separate when written by computer (as a printed character). But there are 28 Arabic characters where each character can be used between two to four ways. i.e.: the shape of character in the beginning of text is differing from the middle or in the last. So, the structure they used for Arabic OCR system is: at first, scan the Arabic documents from right to left. Second, give an image acquisition. Third, preprocessing (which used Binarization to convert image to black and white which make the process on image is easy because it does not cost a lot of measurements and easy to remove noise. In addition smoothing is used to fill the gap between pixels that lost colored), word segmentation, (character fragmentation combination, feature extraction, classification) call it feedback loop, finally recognition results and user interface. The accuracy for this system is 90% with a 20 char/s recognition rate. The similarity between this work and our work is the steps that must be done in an image to get a single character.

But also it differs because these steps are applied on scanned document with white background and black foreground.

Kasmiran *et. al.* [9] a survey presented states a different Arabic optical recognition AOTR used as off-line, and focus the light over the characteristics of Arabic writing. They presented different stages for preprocessing, segmentation, feature extraction classification, post-preprocessing and evaluation methods for different system. In addition, illustrate the popular steps in preprocessing like (1) Binarization, (2) Filtering and smoothing, (3) Thinning, (4) Normalization, (5) Slant correction, and (6) Base-line and Skew Detection. The feature extraction here is for character not for text because authors know there is a text in the image and they jump this step to segmentation step. These features are different from one author to another. In [15] used these features; (1) global transformation, (2) Structural features, (3) statistical features, and (4) template matching and correlation. In [19] used template matching between template of the radicals and character image. In [23] match the histogram of the input characters to those of the templates. The comparative was made between various techniques to give a conclusion that the best algorithm used for binary image is MB2 thinning algorithm. The best author's knowledge, fast and accurate algorithm is the one designed by Amin *et. al.* in [4]. Also, they used decision tree with different masks that give the best result for reliable classification. From this work we can decide or choose the best algorithm and best technique that must be used in AOTR.

Maher *et. al.* in [14] used Dynamic Time Wrapping algorithm which considered as one of the strong approaches to several studies have shown that the OCR based on DTW algorithm provides a very interesting recognition rate especially for large and huge vocabularies. The attractive sides of DTW algorithm is ability recognize properly connected or cursive characters (words or sub words) without prior segmentation. Also, performs the recognition process from within a reference library of isolated characters and owns a very good immunity against noises. But the execution time is very slow and restricts its utilization, because there is a big amount of computing during the recognition process. There are two algorithms used for Arabic recognition

Hidden Markov Model (HMM) and the Dynamic Time Wrapping algorithm DTW. There is a comparative study between these two algorithms. The recommendation is to use HMM to recognize small size vocabulary, otherwise the DTW is one recommended to be used to deal with huge vocabulary such as printed documents library and needs a largest hardware distributed infrastructure. Through the experiments of the authors on around 20000 Arabic words was randomly chosen from high and medium quality documents. The obtained result shows that the recognition rate average is more than 97%, and the segmentation rate average is more than 98% and increases with the size of text font used. The difference here is inserts character in different shapes and styles in HMM to use it in classification, but we need to try this HMM to insert picture for each character and see the result.

Mohieddin *et. al.* in [21] proposed a novel Farsi text detection from video images by corner detection. An edge detector operator in all possible direction, vertical, horizontal, 45, and 135 are extracted. To extract text, some pre-processing is done by dilation and erosion according to the font size. Then corners map are extracted from edges cross point. They used histogram analysis to prevent non text from appear. After that rescale the image to get new corners map. Finally to detect candidate text they use empirical rules analysis. For experimental results they use precision 72.8% and recall 78.54% on 50 images with resolution 720x576. The precision rate is defined as the ratio of correctly detected words to the sum of correctly detected words plus region which actually are not text, but the algorithm detected as text region (false positive Fp) known as wrong detection. Recall rate is defined as the ratio of correctly detected words to the sum of correctly detected words plus region which actually are text, but the algorithm does not detect as text region (false negative Fn) known as missed detection. As a result of some experiments they found the algorithm done on horizontal text not on other direction. This work is related to our work because it works on Arabic text but not on different type of text. Our algorithm can detect variants font style but not on font with word art.

Omar *et. al.* in [24] used customized technique to recognize Arabic alphabets. Also, they illustrated the difficulties faced on Arabic alphabets, and the architecture of system which divided into three parts (preprocessing and line extraction, segmentation and character recognition using neural network to identify the tall and width of characters and to specify the dots and Hamza.). Two important features of Arabic Typed-text are mentioned. The first one is the main line and the other is characters occupies rectangular space in the line. Then they displayed the algorithm and experimental results that appear the effectiveness of the system using different font's type and size with average recognition rate of 87%.

Paul *et. al.* in [25] main idea is how to enhance the bi-tonal images. So, they collect group of degraded Arabic documents (they are depend for document collection to Linguistic Data Consortium LDC, these documents contains Romanize style). They tried to find a manual treatment (solution) to develop a corpus of bi-tonal images.

Volker *et. al.* in [32] discussed the printed and handwritten Arabic words. Evaluation methods to select a best optical character recognition OCR, depends on the best quality. They made a comparison between different published recognition systems for Arabic handwritten. There are different measurements to evaluate the recognition systems such as, testing depends on a large dataset and complexity that solving diverse tests. Second measurement the recognition rate is a global parameter hardly significant for system computer development. Finally, the quality is not enough measure (based on the output recognizer), but the quality of zoning and segmentation into words or characters represents an important feature of recognition system. We take these measurements into account to help us forgive a decision about accuracy and performance for our approach.

All of the above mentioned related work focused on scanned documents with high resolution more than or equal to 200 dpi and used variant techniques and methods, to extract the text from white background with black text, and they give a good result. The aim of the above work is either to create a new document from the degraded document or to measure the similarity of handwritten to achieve certain symptoms like, compare sign on a paper or to recognize the character. But it does not work on other document that contains variant colors, font-size, different orientation and font-style. In the other hand there are many English applications as we can see can be applied on colored background with different shapes for English character and extract English Text from anywhere in the image and printed on a word file. But when we apply it on Arabic image the result either to be unclear or give a result as a picture.

Point out that all related works found in the literature are based on the assumption that the image is known to contain Arabic text, and therefore the purpose of these works is to recognize Arabic characters from the input text. Our focus is on the steps before that identify or existence text in the image. However, most of the techniques used in these approaches can be utilized in our work.

2.3 Optical Character Recognition/Optical Text Recognition Applications

There are many applications depend on OCR techniques and text detection to extract text from complex image which are divided into two parts: the first one work with English text and the other work with Arabic text. Some applications work with English (i.e. MobiReader and ABBYY) is needed to install in computer and the other work free on a web. It can be applied for two ways: first way is select a specific area from image to extract the text either into word or on a web page ...etc. The second one is choosing a homogenous image with complex or white background to detect and identify the area which contains text, then extract the text into word or other types of files. The second part is an application or software includes Arabic as a choice and detects the area which contains text but the result is not in an efficient way like OCR-TextScan and Readiris. On the other hand there is an application based on select a specific region that contains a text to recognize and extract Arabic characters, some of them is a free on-line application like free Online OCR, and the other is a tool comes with windows or office like OneNote 2007, and Microsoft Office Document Imaging.

MobiReader Biz+ (Business Card OCR Reader) is a business card recognition application developed by DIOTEK. It has many facilitates one of them is OCR technology. The image captured with an iPhone which can be converted into text and stored by using OCR. According the virtual business card holder function could make a phone call by selecting a telephone number in the Business Card Holder and send an email message or SMS [50].

Another application/software is ABBYY FineReader 11 professional Edition refer to Appendix A for exclusive details, which export or saving to an extend application via one page at a time [47]. This application tends to work on English and on Arabic pages. It gives a good result in English but in Arabic is misunderstand because does not support Arabic, although the description of the software claims it support Hebrew, Farsi and Arabic.

Another program is: OCR-TextScan 2 Word 1.0.lnk program "can easily scan paper documents. The program now tries to find the text information out of the scanned picture and to save it as word file or text file. Still have to correct the texts but you save a lot of time compared with complete retyping of the text. Text in the most used fonts can be recognized. Can start the OCR process over command line for large amounts of scanned files. Only have to provide the program with the name of the picture file and the output RTF/DOC file". [This information written in the user manual comes with the program]. But when tried this program to open an image and extract the text, can't give any clear information neither Arabic nor English.

Also, we try the Readiris Pro 8.0, "With Readiris 8.0, the OCR software detects columns in the document and can recreate them in the output file. Scan a columned document produces a Word document with editable columns. By editing the text, the text “flows” naturally from one column to another! Recognizing a color page can now easily take a few seconds less - and Readiris was already the fastest OCR package on the market!" [This information written in the user manual comes with the program], but it does not provide any result.

Other free applications in [40],[41] and [45], such as Microsoft office OneNote 2007, and Microsoft Office Document Imaging from Microsoft Office Tool, these are depend on selecting the part of text then paste it on any document but does not work correct in Arabic.

From the previous related works we can summarized as follows:

- Most Arabic related works interested in segmentation process to increase the performance and accuracy either in handwritten or in typed characters.
- Languages rather than Arabic solve the problem of extracting text from graphics image either automatic or by selecting the region of text.
- There is an OCR application success to extract automatically English and other languages rather than Arabic from graphics image.

- Arabic research late one step from other languages in this area of interest.

The next chapter presents the proposed solution to solve the problem of extracting text from graphics image and removing other things. This solution works as filter that could be used as previous step for any work uses text processing.

Chapter Three

Research Methodology and Techniques

In this chapter we will present two sections. The first one is our technique which presents automatic text/word segmentation and solves many problems in colored image until reach the final goal. This goal is creating filter to extract text form colored images. The second section presents the implementation structure for the main functions used to build the filter. But before discussing these sections, we will present a research design and methodology to get brief details.

3.1: Research Methodology

Our research will be composed of three main stages: first, define most suitable set of features for Arabic. Second, we will develop our Arabic text recognition model. Third, evaluate our newly developed model and integrate it into an OCR system to evaluate its performance within the overall OCR process. These stages and their details are depicted in *figure 4*, and are further explained in the following two subsections.

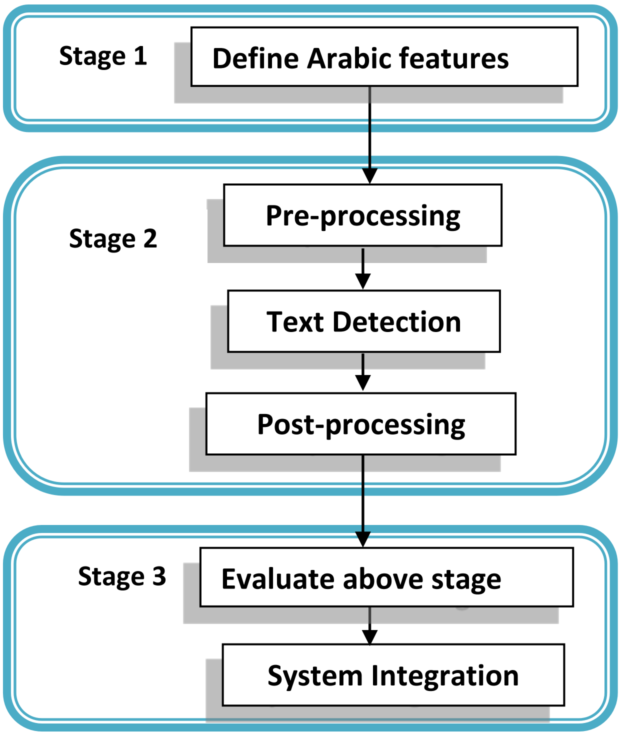


Figure 3. 1: the model for research design

Stage 1: Define most suitable set of features for Arabic

In this stage we try the most suitable features for Arabic. Like, (1) seeking process must be done from right to left, (2) base-line to identify the orientation of Arabic text, (3) to decide if this region is an Arabic text or not may use some methods to calculate the number of corners, vertical, horizontal, and curved lines.

Stage 2: Arabic Text Recognition

The Arabic text recognition stage encompasses three steps: 1) pre-processing, 2) text detection stage, and 3) post processing. These stages are explained as follows:

Pre-processing

In this stage the following pre-processing operations are performed on the input image to improve the detection efficiency

- Extract intensity from color image
- Intensity enhancement
- Noise removal
- Thresholding (binarization)

Text Detection

This stage contains the core processing operations that we will develop.

- The outcome of this stage is to localize the sub regions which contain Arabic text. Localization can be given as a bounding box, counter, or centroid.
- Our technique's performance will be evaluated visually based on the percentage of text regions it is successfully able to detect. Also, make a correlation function that checks how much the output result and the exact output are equals.

Post-processing

- Clean extracted text sub image from any remaining background clutter.
- Rescale and resize the image to be suitable for OCR segmentation (segmenting words into sub-words), and recognition (recognizing each letter separately).

Stage 3: Evaluate integrating our module into other applications

This stage involves two steps: first, evaluate our model by showing the rate of correct text localization from all the contents of graphics image and this step might be done manually. Second, search for available OCR systems which can integrate with our module. Candidate systems are:

- ABBYY: is considered one of the top 5 programs in text recognition.
- Gimp: a free and open source image manipulation program.

This requires a thorough analysis of these systems and a process of tailoring our module to integrate with the system of choice.

Our algorithm for extracting text written in horizontal way from colored images which prepared by PowerPoint, Photoshop or any application capable to design an image, passed through several variations before reaching the final goal. In this research we propose a new technique for automatically detecting text in graphics documents, and preparing them for OCR processing. Our detection approach is based on finding regions in the input image with high density of text features.

When we started work in this research we thought to get these features by collecting information about angles, lines, and curves. To help us for detection regions are subsequently processed to enhance the likelihood for successful character recognition by existing OCR techniques. But through our research we could not find any method to give these text features.

So, our approach to solve the problem of recognizing text from colored images is depending on generates a set of rules based on comparisons between the most important features of text than graphic. Such as computes a ratio of item area (number of black pixels) over the segment area to take a small factor that separate between text and other graphics. The other feature is Arabic text was written just on one line which we call it 'Base Line', where the density of black pixels larger clear, rather than English text which was written on four lines. Finally compute the maximum peak for adjacent black pixels in the Base Line to know if it is English or Arabic. Because we claim that Arabic text has a long horizontal adjacent line rather than English which is written as separate characters.

So that to solve the problem there are three steps. These steps are 1) preprocessing, 2) text detection, and 3) post processing; and their implementation will be discussed in the next sub sections.

3.2: Techniques

Based on the set of Arabic features, different techniques are discussed to select the best one which provides an acceptable performance and results.

3.2.1: Preprocessing

In this stage we are making some preparations on the original colored image to be ready to text localization stage. This preparation revolves around remove some noise from background image. Also make the background to have a uniform color. Finally convert the original image to black and white image.

The first preparation is converting colored image to gray level image. Then we need to perform an edge map. There are many filters such as *Canny*, *Sobel* and *Prewitt*. But any one does not enough to get a suitable amount of unwanted details and to represent the most foregrounds. So, after searching into many filters we decide to use these two filters (***wiener2*** and ***rangefilt*** respectively). A graythresh**,** wiener2 and rangefilt are MATLAB build in functions (see figure 3.9).

"Wiener2 lowpass-filters a grayscale image that has been degraded by constant power additive noise. wiener2 uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. The additive noise (Gaussian white noise) power is assumed to be noise."

"Rangefilt determines the center element of the neighborhood by floor((size(NHOOD) + 1)/2). "

Our decision is coming from two reasons. The first reason is to remove some noises like graphics from the origin image by keep it the most amounts of elements in the image. The second reason is to perform image with just two colors (black background and white foreground) to help us to deal with the variance background color in the same origin image. This situation is required to apply other techniques like segmentation process and skeleton (thinning) process.

After that we use ***Graythresh***. It is an automatic threshold to find a suitable threshold that associate for each image separately. "The graythresh function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels."

Finally, the combination between filter results and automatic threshold as describe in algorithm 3.1 is to create black and white image (Binary image). Provided that the background should be white and the foreground should be black to look like an A4 document.

Algorithm 3.1: threshold for edge map

```
Input: colored image I
Output: binary image
    1. Convert I to grayscale image
    2. K = image after applying filter to give a uniform background
       and foreground color, and to remove noise
    3. Level = the value of threshold for K
    4. BW = convert grayscale image to black white (K, Level)
```

The segmentation process with edge detection is very comfortable, because it makes the background for any colored image to be black and the foreground to be white as a uniform color. Then one technique can be used to make the process of segmentation for any RGB image regardless of its content with high speed. But without edge detection, the RGB image should be converted to binary image through the conversion to grayscale directly, which may produces a different variation of background and foreground between black and white. So, we need to apply different techniques based on exchange between black and white to allow us to use a segmentation process which decreases the execution speed. In the next section we propose the adopted segmentation process to identify the localization of objects.

3.2.2: Text Detection

This section is the most important point in our research and it refers to the primary aim to reach final goal.

Three steps are discussed namely, text localization, collect information, and text type recognition.

3.2.2.1: Text Localization Method

This section depends on segmented the image and extract all items that are included after converted the image from RGB to binary, then identify the localization of Arabic text.

It is important to say this step is most difficult because it takes a lot of time and variant trials to find a way for solving this problem. In the beginning we thought to create a two dimensions array that holds information for the entire binary image. The information as follows line number, number of white pixels, and number of black pixels for each line. Then we used this information as input data in order to enable us to deal with image. So, we apply three stages:

Stage 1: the first one is segment the image horizontally (row by row) considering that the text was written in horizontal way. A long horizontal white line is calculated i.e.: number of white pixels equal image width. But this way faces a problem which is the segmented image might include more than one item like picture with another picture, or picture with text (the text here could be an English or Arabic). Whatever, go to second stage.

Stage 2: we developed a code to segment a sub-image vertically to separate between items in one sub-image. By saved an information for each sub-image separately, but this time read an image column by column to have a two dimension array includes column number, number of white pixels, and number of black pixels. To find long vertical white pixels equal to high of sub-image. But another time this way face a problem with sub-image includes more than one item have an interlacing between of them, the interlacing means that, between the two items there are no white column to allow us to apply vertical separation process. So, we go to next stage.

Stage 3: *a connected label algorithm* is used to label close items with same number depending on identify a small ratio of distance. To be easy to know the coordinates of each item and then separated as rectangle form. The value of distance is chosen by select and test different values until we found the nearest value for extracting process which gives a nearest connected item in one line. If minimize the distance value as much as possible can get each item alone according the long of the space between items. So, we found a **'bwdist ()'** method, to identify the distance value and choose it to be four, and **'bwlabel ()'** method, which helps us to separate between items in the same sub-image based on that distance value, by giving a separate number for each connected item. After that we use the labels for each item to take the minimum and maximum coordinates [row and column] (x1, y1) and (x2, y2) to identify a rectangle or square shapes around the object. As an example (see figure 3.2, 3.3, 3.4 and 3.5).

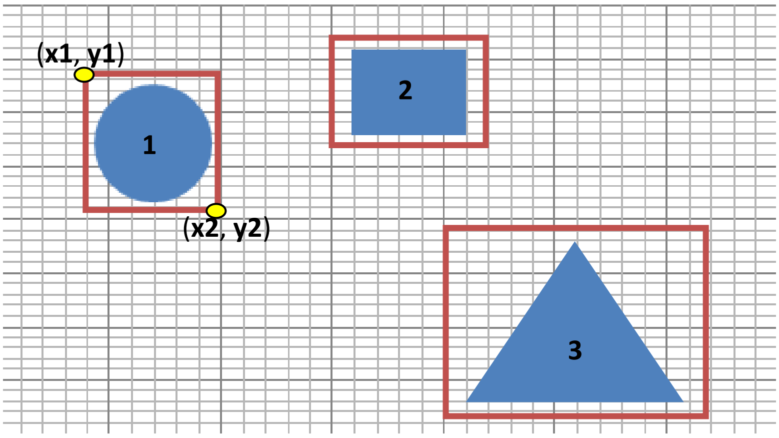


Figure 3. 2: example to show how identify a rectangle shape over an items

		1	1		
	1	1	1	1	
	1	1	1	1	
	1	1	1	1	
		1	1		

Figure 3 .3: circle item with label one

	2	2	2	2	2	2	
	2	2	2	2	2	2	
	2	2	2	2	2	2	
	2	2	2	2	2	2	
	2	2	2	2	2	2	

Figure 3 .4: rectangle item with label two

					3					
				3	3	3				
			3	3	3	3	3			
		3	3	3	3	3	3	3		
	3	3	3	3	3	3	3	3	3	

Figure 3 .5: triangle item with label three

According try the above stages in that order, in shuffle with each other, or using connected label method alone. The number of segmented items with three stages is better than using connected label alone. For example: if the image contains a border and the item is very close to that border, when we increase the value of distance in a small amount it is effect on separate process and regards them as a single item, thus minimize the number of sub-images that will enter to final stage which examining if the item is text or not.

In spite of this, when we use the connected label method without any addition stage that we talked about, it gives most of sub-items that are located in the entire origin image. So, we were forced to use it more than one time to get each item alone. We have to, because this method works only when the background is black. So, for this reason there are two cases. The first one is, when we convert the entire image to binary image all pixels in the background is white (i.e.: contains ones). So, we need altered to zeros to able to put labels for each item in that object. The second case is, after convert to binary image there are variations of background between black and white. Therefore the sub-object that has white background does not work with the connected label method in a proper manner. So, we exchange the background to be black, then again use connected label. This led to effect of the execution speed to become slower.

To explain the reason for using connected label method twice, we will strike an example for more declaration.

Look at (figure 3.6), picture with label 1 is the original image with colored background. Picture with label 2 is black and white image. Picture with label 3 is a centroid image which illustrates the blue star in the middle. This meaning all the contents is one item. And picture with label 4 is a result of applying a connected label method. As a result of this the segmentation process does not work correctly to separate the elements in the original image has colored background.

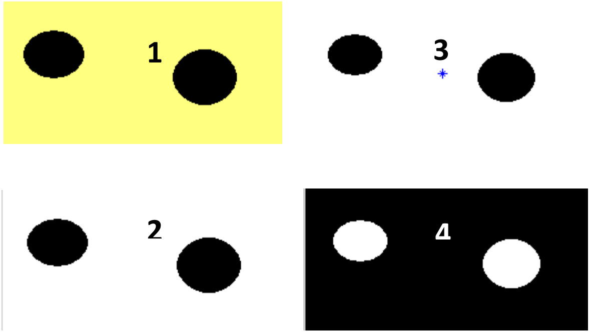


Figure 3. 6: example 1, Apply connected label on colored background

For more declaration look at (figure 3.7). We have in part 1 image with two different backgrounds, one of them is white. Label 2 is a black white image. Label 3 contains two centroid stars one of them is exactly in the centered of item, but the other is outside the item. That means there are two items in the original image one of them is correct and the other is not. So, when we apply connected label, it works only in the part of binary image have black background as you see in part 4. Thus the method recognized only one item. For this point we need to exchange the partition which has a white background to be black and again apply connected label method for this part. This case exactly needs to apply connected label method twice.

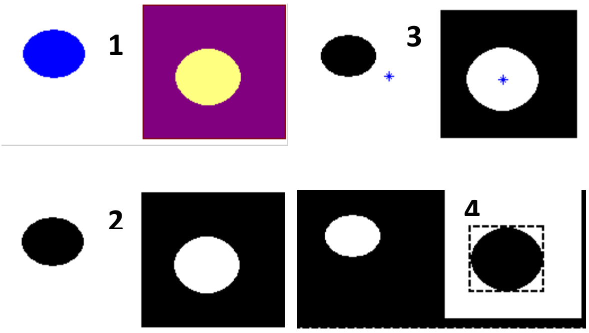


Figure 3. 7: example 2, Apply connected label on colored and white background

In (figure 3.8), we can see the original image in part 1 has a black background and two white items. In part 3 we have two correct centroid stars in the middle of each item. As a result of this we get a correct final result in part 4. That has two dashed square boxes around the items. So, the localization process for each item in original image is done.

Note: the background in part 4 in (figures 3.7 and 3.8) are white because we exchange the color between background and entire elements to display the rectangle shape around each item.

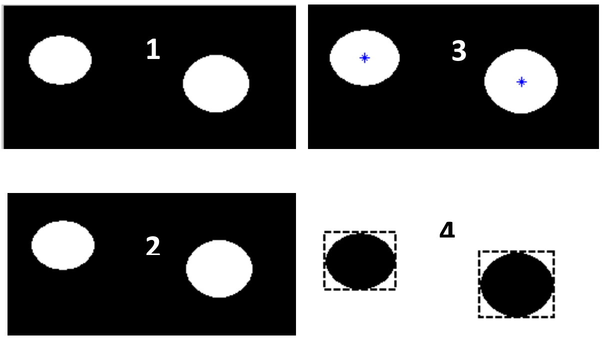


Figure 3. 8: example 3, Apply connected label on black background

The new idea is depending on connected label only with small distance (we assume it equal to one) including information about each item. This information is **maximum row** and **column**, and **minimum row** and **column,** which are preserved in a two dimension array with two additional fields. These fields are **object number** and **label.** An ascending sort is applied to the content of the two dimension array depending on the value of minimum row *see algorithm 3.2*. **Object number** holds the value of item order that presents by connected label method. **Label** initialized with zeros and use it as classification field to put a new label for the same item with the nearest minimum row, minimum column and maximum row. These values depending on how much the content are closed on each image. So, according to the label value we make groups or regions of items that verify pervious conditions.

These conditions come from the computation of the difference between the minimum rows. The differences saved in one dimension array. Items in the same row have minimum difference and if the difference has large value means that is a pointer to start a new row. So, to know the difference is small or big we compute the mean of the differences values. The mean value is used as a threshold to identify the items in the same row, and new labels are provided if the difference is larger than the mean value *see algorithm 3.3 and 3.4.*

But we face a problem in some items. The problem is, some items in the same minimum row should be in the same region but perform more than one region or group. This problem was appearing because items appear in random order. So, for this reason we apply another ascending sort but this time on the value of minimum column. This sort applies on the items belong to the same label. Also, compute the differences between the maximum columns and minimum columns then find the mean to know if the item is near enough to the previous item or not. So that we decide to give a new label for this item or give it the same label for the previous one. After that, sort labels in ascending order to find coordinates in an easy way *see algorithm 3.5.*

Finally, take the minimum and maximum row and column for all items have the same label to perform new selected region. Each value stores in one dimension array. Then take these regions to know is it text or not *see algorithm 3.6 and 3.7.* To show the results see (figure 3.9 and 3.10).

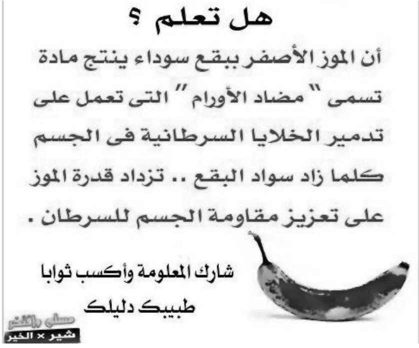
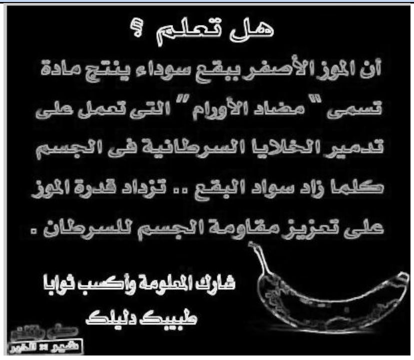
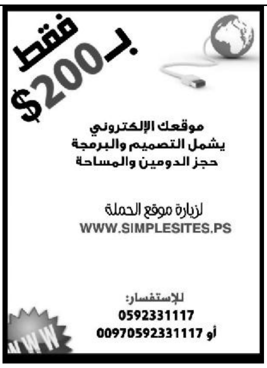
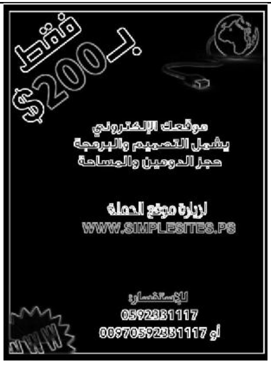
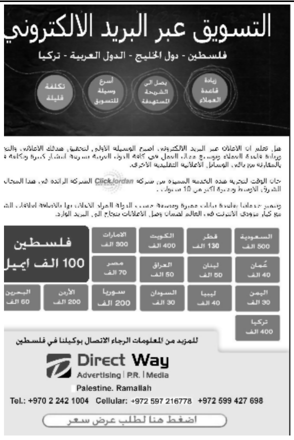
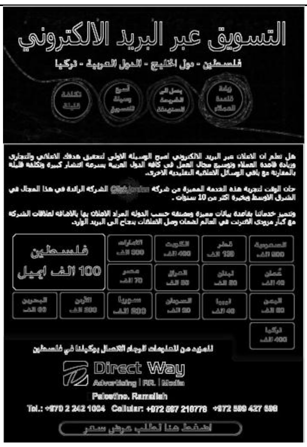
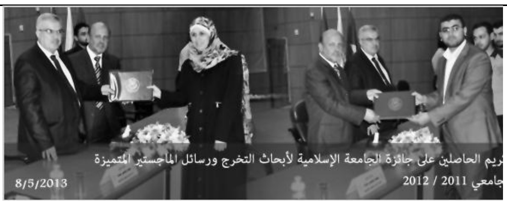
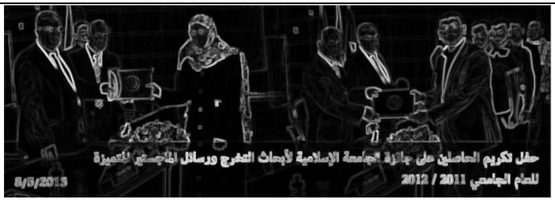
Wiener2	Rangefilt
 هل تعلم ؟ أن الموز الأصفر ببقع سوداء ينتج مادة تسمى " مضاد الأورام " التى تعمل على تدمير الخلايا السرطانية فى الجسم كلما زاد سواد البقع .. تزداد قدرة الموز على تعزيز مقاومة الجسم للسرطان . شارك المعلومة وأكسب ثوابا طبيبك دليلك	 هل تعلم ؟ أن الموز الأصفر ببقع سوداء ينتج مادة تسمى " مضاد الأورام " التى تعمل على تدمير الخلايا السرطانية فى الجسم كلما زاد سواد البقع .. تزداد قدرة الموز على تعزيز مقاومة الجسم للسرطان . شارك المعلومة وأكسب ثوابا طبيبك دليلك
 بـ200$ فقط موقعك الإلكتروني يشمل التصميم والبرمجة حجز الدومين والمساحة لزيارة موقع الحملة WWW.SIMPLESITES.PS للإستفسار: 0592331117 00970592331117 أو	 بـ200$ فقط موقعك الإلكتروني يشمل التصميم والبرمجة حجز الدومين والمساحة لزيارة موقع الحملة WWW.SIMPLESITES.PS للإستفسار: 0592331117 00970592331117 أو
 التسويق عبر البريد الالكتروني فلسطين - دول الخليج - الدول العربية - تركيا فلسطين 100 الف ايميل Direct Way Advertising | PR. | Media Palestine. Ramallah Tel.: +970 2 242 1004 Cellular: +972 597 216778 +972 599 427 698 اضغط هنا لطلب عرض سعر	 التسويق عبر البريد الالكتروني فلسطين - دول الخليج - الدول العربية - تركيا فلسطين 100 الف ايميل Direct Way Advertising | PR. | Media Palestine. Ramallah اضغط هنا لطلب عرض سعر
 ريم الحاصلين على جائزة الجامعة الإسلامية لأبحاث التخرج ورسائل الماجستير المتميزة جامعي 2011 / 2012 8/5/2013	 حفل تكريم الحاصلين على جائزة الجامعة الإسلامية لأبحاث التخرج ورسائل الماجستير المتميزة للعام الجامعي 2011 / 2012 8/5/2013

Figure 3 .9: show samples after applying wiener2 and rangefilt filters

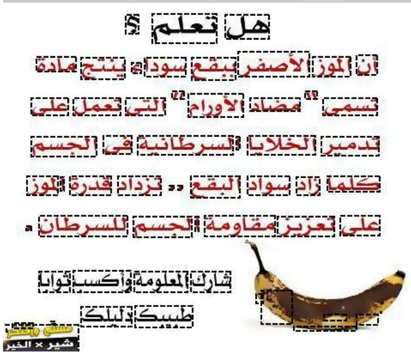
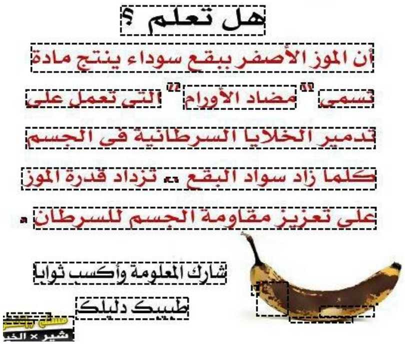
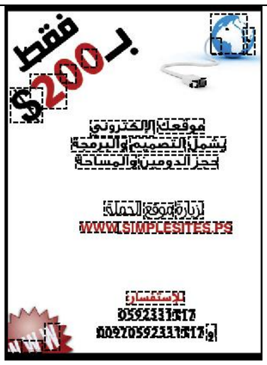
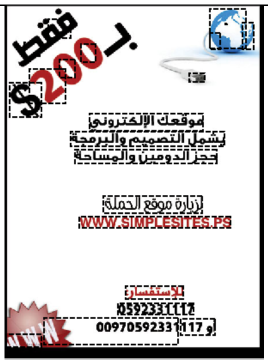
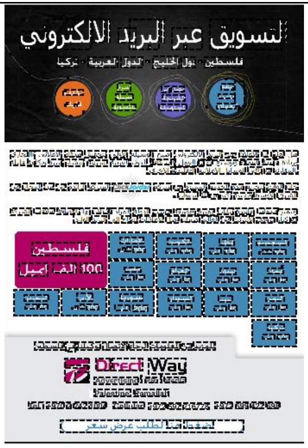
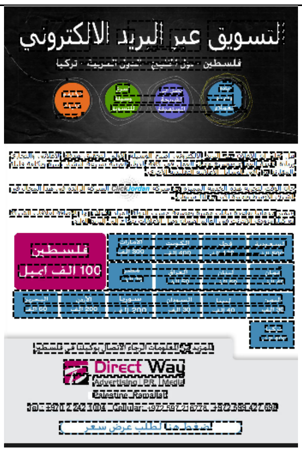
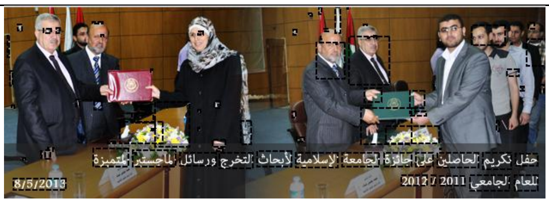
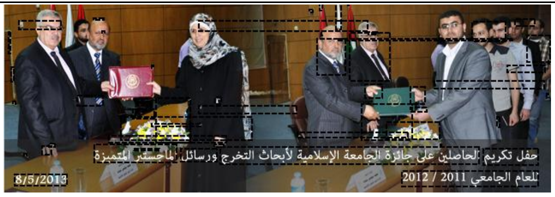
Connected label segmentation	Our segmentation
	
	
	
	

Figure 3 .10: example to show localization result

Algorithm 3.2: Localization process step 1

(Collect information)

Input: Binary image BW, I

Output 1: localized each item in I via dashed rectangle shape

Output 2: ascending sort object_array by minimum row

1. bw3 is a computation distance of BW
2. find the labels for each elements in bw3
3. Put any other elements in the image like background to zeros.
4. Define 2D object_array with six items to save element number, maximum row, maximum column, minimum row, minimum column, and last one to save new label.
5. Draw rectangle shape with these parameter (Mnc, Mnr, width, height, line width, line style)
6. Sort 2D object_array in ascending order depending on min_row.

Algorithm 3.3: compute difference and mean for minimum row

Input 1: ascending sort object_array by minimum row use algorithm 3.2

Input 2: z number of items

Input 3: kind variable, it is maybe max or min row or col.

Output 1: mean for input 1

Output 2: difference_array1 vector for input1

1. Define Difference__array by length z
2. Each index in Difference_array contains the value of difference between two adjacent items in one column of max/min row/col in object-array.
3. Now compute summation for all items in Difference_array vector.
4. Find the average and get the floor value.

Algorithm 3.4: Localization process step 2

(give label for each new row)

Input 1: ascending sort object_array by minimum row use algorithm 3.2

Input 2: difference array of min row (see algorithm 3.3)

Input 3: average of differences of min row (see algorithm 3.3)

Output 1: ascending sort object_array by minimum column

1. Put 1 into the label of first item in Object_array
2. Check the value in difference_array according the minimum row to be greater than the average of difference_array
   - If the check is true then increment the label value
   - The rest items labeled with zero.
   - Give the new label value to the correct check item and so on
   - This operation identify the beginning item in each row
   - Save the place of each new label and use it as a range in sort by min column
3. Sort (object-array, Obj, Xl, min_col) → sort elements in range of each label in ascending order depending on min_column.

Algorithm 3.5: Localization process step 3

(give labels for the rest of items in object_array)

```
Input 1: ascending sort object_array by minimum row and column use
         algorithm 3.4.
Input 2: Avgmaxrow is average of max row (see algorithm 3.3)
Input 3: Avgmincol is average of min column (see algorithm 3.3)
Output 1: sort object_array by labels in ascending order

    1. In this algorithm need to identify if each item belongs to
       the new label or it should get a new label value according
       the distance between each item in one row
         i. A is a pointer in the first item
        ii. B is another pointer to the next item
       iii. Check the difference of maximum row between the
            two pointers is less than or equal the average of
            maximum row then
              1. Check the difference of minimum column
                 between the two pointers is less than or equal
                 the average of minimum column then
              2. B get the same label of A
              3. Else increment the label value
              4. Label of B equal new label value
        iv. Else increment the label value
         v. Label of B equal new label value.
    2. Sort (object-array, label) → sort all elements in
       object_array by label in ascending order.
```

Algorithm 3.6: Localization process step 4

(find minimum and maximum row and column after new label classification)

Input 1: ascending sort object_array by label use algorithm 3.5.

Output 1: start_min_row vector is a vector that contains all minimum rows for after classification labels

Output 2: start_min_col vector is a vector that contains all minimum columns after new classification labels

Output 3: end_max_row vector is a vector that contains all maximum rows after new classification labels

Output 4: end_max_col vector is a vector that contains all maximum columns after new classification labels

1. Each vector in output 1, 2, 3, and 4 initialized with values of first item in 2D object_array.
2. Make a search for each label to find minimum and maximum row and column to identify the area of new label.

Algorithm 3.7: Localization process step 5

(select text region)

```
Input 1: start_min_row vector is a vector that contains all minimum rows for after classification labels use algorithm 3.6.
Input 2: start_min_col vector is a vector that contains all minimum columns after new classification labels use algorithm 3.6.
Input 3: end_max_row vector is a vector that contains all maximum rows after new classification labels use algorithm 3.6.
Input 4: end_max_col vector is a vector that contains all maximum columns after new classification labels use algorithm 3.6.
Output 1: vector of text region crop
Output 2: obj,region number
Output 3, 4, 5, 6:start_min_row, start_min_col, end_max_row, end_max_col vectors

    1. Compute width and height for each new object
    2. draw rectangle a round new object
    3. crop the new object to check if it is a text or not
```

To do previous steps we need to collect some information. The technique we use it is described in next sub section.

3.1.2.2: Collect Information

This information is necessary to deal with crop of images in a proper way. Such as to find the larger row that include maximum black pixels (i.e.: Base line). To find maximum peak value, is calculated the frequency of smaller connected black pixels in the base line. Values of peak and base line are using to build a set of formulas and rules to get final goal.

For collecting information from sub-images we will use a simple technique. This technique is read the image line-by-line. Then check each pixel in each line if it is black or white. After that, pick the number of each color in separate variables.

So, we create a two dimension array contains three attributes (column) and Z row, to connect each number of colors in each line. Where, Z is the number of rows in the sub-image. These attributes are row-number, number-of-white-pixels, and number-of-black-pixels. *See algorithm 3.8.*

This algorithm is used because we need to exchange white to black background to make connected label algorithm worked in an efficient manner as we explained in *section 3.2.2.1.* So, there are some region of crop-images still has a black background not all of them. So, we need this algorithm to get information. For example, check the number of black and white pixels. If black is greater than white in the region of crop-image then you can swap the color for this region image only not for all sub-images to be white background and so on.

Another usage is finding the maximum row that has large black pixels. Also, find the peak for the frequency of maximum small black pixels, to know if the statement is Arabic or English. You can see more usage for this algorithm in next section.

Algorithm 3.8: collect information

Input: BW3 region of Binary image crop

Output: full-info-matrix is a Tow dimension array with three column

```
find the size for each crop object
b=count black pixel in each row
w=count white pixel in each row
save row number into first column of full-info-matrix
save b  into full-info-matrix into second column
save w  into full-info-matrix into third column
```

The above mentioned algorithm keeps information for all regions of crop-images which are stored in full-info-matrix array.

Now we will illustrate the ***Calculation Arabic Base Line algorithm***. This algorithm based on make a comparison between the amounts of black pixels for each row in the sub-image. The amounts of black pixels saved in column three in full-info-img array. Assume the first value in full-info-img is a maximum value and name it ***maxm***. Also assume ***Base*** variable is the first row. Then start the search process from the second row to last row. If maxm less than the value of black pixels in the next row, change the maxm value to the new value of the next row, *see algorithm 3.9.*

Algorithm 3.9: Calculation Arabic Base Line

Input: two dimension array full-info-img use algorithm 3.8.

Output: The Base Line Number

```
1. make a search to find row that contains maximum black pixels
2. compute half line
3. check if this row is below the half line
```

Algorithm 3.10: find lengths of adjacent sequence black pixels in Base line

Input: Base line for BW3 region of Binary image crop use algorithm 3.9.

Output: descending Bline vector, this vector contains length of adjacent sequence black pixels in base line

```
1. find the size of BW3
2. Initialize Bline() vector by zero. The length of Bline is equal
   the width of crop image.
3. For all pixels in the crop image
   a. Check if the two adjacent pixels are black, then count
      black pixels in sumb variable.
   b. If the two adjacent pixels are different from each
      other then,
        ▪ Save sumb in the first index in Bline
        ▪ Increment the index of Bline by one
        ▪ Reinitialize sumb by one.
4. Sort Bline vector in descending order.
```

Algorithm 3.11: find accumulator for lengths of adjacent sequence black pixels in Base line

Input 1: descending Bline vector, this vector contains length of adjacent sequence black pixels in base line use algorithm 3.10.

Input 2: x value, where x is length of Bline vector

Output: accum(x, 2) is a two dimension accumulator, first column is the length value and the second column is how many times this value redundant in Bline vector

1. After sort Bline vector, all equalize values are come in front of to each other so,
2. Initialize accum with zero
3. For all elements in Bline vector do the following
    - Count the equalize values and save it into the accum() in index two,
    - Save the actual value into the accum() in index one

Algorithm 3.12: find peak and its position

Input 1: accum(x, 2) is a two dimension accumulator, first column is the length value and the second column is how many times this value redundant in Bline vector use algorithm 3.11.

Input 2: bc value, where bc is length of accum two dimension array

Output 1: maximum peak value

Output 2: C, the position of maximum peak

```
1. Initialize peak by 1
2. Initialize C by 1
3. For all elements in accum()
      ▪ search for the maximum peak value and save it in peak variable
      ▪ store its position in C variable
```

3.2.2.3: Text Typed Recognition

This section is the final step in text detection. It is about how we can know if this part of image [*I mean here the region of image*s_crop] is a text or not? Especially here we make a search only for Arabic text. For this reason we focus on the most important feature marked Arabic script is ***Base-Line***.

Firstly to distinguish between text and other shapes or figures in the origin image, we compute the region area using existence method ***bwarea().*** This function returns the area of a binary image. The area is a measure of the size of the foreground of the image. Then divide the bwarea over the region area to give a specific ratio for each region.

Secondly measure the Base line to display different Arabic font size and style. Base-Line is the horizontal (*not vertical, not diagonal and not curved*) straight lines which use it for writing on Arabic papers. Over this line there are characters are written either above or below the line according to the script of each Arabic character. So, the intensity of black pixels in this line is very clear and you can call it a winner row. In addition Arabic base line has a big amount of density of color after apply thinning over the region of crop_image. Intuitively, the base line comes in specific place below the half of the region of crop_image not in the above. Therefore, any row exists in the upper half or any place rather than the winner row is not acceptable. Thus we minimize the number of region of crop_images that we will be checked. The check process is only restricted for text not for other shapes. Sometimes this feature removes some of English text.

Thirdly, to distinct between English and Arabic, we use feature of English script on Base line. This feature is the English character either small or capital letter wroteon Base line after thinning has maximum peak for one black pixel. *See algorithm 3.10, 3.11 and 3.12*. Thus, some English text is removed from the output result. You can see the results in *chapter 4*.

In the next sub sections, we illustrate the **calculation Arabic Base line ratio** algorithm. This algorithm helps us to display text with different font size and style. **Set of rules** algorithm shows the sequence steps to get result that use as input to OCR system.

3.2.2.3.1: Find Base Line Ratio

We used the length of height for each image_crop to measure the ratio place for Arabic base line as follows:

Algorithm 3.13: find half line and Base line ratio

```
Input 1: imx, is the height region crop
Input 2: Base value for the region crop use algorithm 3.9.
Output 1: halfline, the value of half line
Output 2: ratio, range that maybe found place of base line in the region
          crop

    1. halfline is the floor of the middle row of crop image
    2. defhalf is the distance between the height of crop image
       and halfine.
    3. defBase is the distance between the height of crop image
       and Base line.
    4.   If (defhalf < defBase) then
           ▪ ratio is the floor of  defhalf over defBase
    5.   Else
           ▪ ratio is the floor of defBase over defhalf
    6. Finally, ratio is equal ratio multiply by Base
```

3.2.2.3.2: Set of Rules

This section is the last step in our work. It is a compilation of previous values. We will use the most nearest values in a collection of rules. These rules can identify the largest amount of Arabic region of images crop from the set region images crop. Now, as we see from the beginning of this research we want to display only the Arabic text images. So, in next algorithm we are explaining the rules as much as possible.

We have some restrictions in our work. The *first* one is the height of sub-image does not exceed than seventy-five pixels to ignore the largest shape. This restriction is used from the beginning before saving the items in an object array to establish new label.

The second restriction is using fixed number in the conditions. These numbers identify by manual experiment.

After that we minimize as much as possible the amount of region images that not related to text. Now we put rules that related to exclude the Arabic region images from the rest of regions as: 1) compute the ratio of foreground area to the total area of region to give small semi identical ratio to prevent display graphics on the output result. 2) find the position of Base line, the position of maximum peak and its value to prevent display English text in the output result. The output result is an empty image has the same size of the original image but with white background. The final result is formed to be an input to OCR system. *See algorithm 3.14.*

Algorithm 3.14: Set of Rules

```
Input 1: objarea, is the ratio of foreground area to the region area
Input 2: Base, the value of base line
Input 3: halfline, the value of half line
Input 4: ratio, the range could be found the base line below the half line
Input 5: accum two dimension array, includes in first column the times of
         strait line of black pixels in the base line are redundant
Input 6: val, the value in the second column of accum array
Input 7: C, the position in accum that contain maximum peak value
Input 8: start_min_row, start_min_col, end_max_row, end_max_col
```

```
        vectors, these vectors contains the origin coordinates from
        the origin image
Input 9: BW3, thinning image crop
Input 10: I2, blank BW image with white background and has the same
          size of original RGB image
Output: I2 output result image, this include Arabic text with some
        unwanted data
  Apply algorithm 3.7 (select text region) to return obj,
  start_min_row, start_min_col, end_max_row, and end_max_col.
  Convert white background to black
  BW3 = thinning to image_crop
  Farea = Find foreground area
  Imagearea = Find image_crop area (width * height)
  Objarea = Farea / Imagearea
  Apply algorithm 3.13 to find halfline and ratio
  Apply algorithm 3.11 to find val and C
  Size (I2)=size (I)   →   I is the origin image
  I2 = 1   →   i.e.: I2 has white background
  if (Objarea >= 0.01)and(Objarea <= 0.16)
      if (Base >= halfine) and (Base <= hafline + ratio)
          if((accum (C, 1) >= 1) and (val >= 3))
          Copy the segmentation part from the region image crop to its
          position in I2 output image.
          End
      End
  End
```

All of the previous algorithms are integrated into one method namely preprocessing ***().*** Now in the next section we will describe the ***postprocessing ()*** method.

3.2.3: Post Processing

In this section we have two post processing depending on the output result. One of them for OCR system with white background and black foreground but size differs from image to other. The other output is for users to be easy to show the correct text recognition. The text marked with red color on the RGB image with only black and white colors as a copy from the RGB image.

3.2.3.1: Result Image for OCR

After finish the step of text detection the output image has some items, shapes and English text should be not appear on the result. So we decide to take the result and inserts on more process as a trial step to give OCR system a clear image result as much as possible.

Thus for this we take the output image I2 from *algorithm 3.2.2.3.2.1.* Then make segmentation on thinning image with big distance to select long text region. Then put some rules that calculate the ratio of black pixels to white pixels. We have to, because ratio of black pixels after thinning in Arabic text is less than white pixels in the shapes and English text. These ratios are selecting by manual experiments give good results. But we go in a close end, because if put excess conditions, maybe lose some items we need it to appear. *See algorithm 3.15.*

Algorithm 3.15: OCR Post Processing

Input: I2 output result image, this include Arabic text with some unwanted data use algorithm 3.13

Output: I3 output image to OCR system

```
1.  Apply algorithm 3.2 and return clip region maxr, maxc, minr, and
    minc. but with change the distance with 4 and without use
    objectarray and ascending sort
2.  [dx dy] = size(clip)
3.  BWarea = foreground object area
4.  cliparea = dx * dy  → (width * height)
5.  blackratio = 2/3 * cliparea
6.  countwhiteratioarea=countwhite / cliparea
7.  countblackratioarea=countblack / cliparea
8.  blackoverwhite=countblack / countwhite
9.  if (countblackratioarea <= 0.28)
10.     if (BWarea > 30)
11.         if(blackratio / BWarea >= 2.5)
12.             If (blackoverwhite < 0.251)
18.         Copy the segmentation part from the region image crop to its
            position in I2 output image.
13.           End
14.         End
15.     End
16. End
```

3.2.3.2: Result Image for Display

This result is just used as a displaying image that selects text region on original image. Therefore, the appropriate way is take a copy from RGB image and converted to black and white image. This helps to present red color clearly. Then multiply red white image with black white image.

Algorithm 3.16: Show Post Processing

```
Input: I, is a colored image
Output: Img, is output image for show
    1.  I2 = convert to binary of I
    2.  I3 =  preprocessing ( I )  →      I3 and I4 is binary image
    3.  I4 = postprocessing ( I3 ) ↗
    4.  Swap between black and white in I4
    5.  Now apply thicken skeleton on I4
    6.  Swap again between black and white in I4
    7.  rgb = Convert I4 image to RGB image
    8.  I2 = Convert I2 image to RGB image
    9.  Change foreground color in I4 to red color
    10. Img = multiply I2 with rgb
```

This is our idea to solve the problem of extracting Arabic text images from the colored background image (for more details refer to *Appendix B).*

3.3: Implementation

Our work is implemented by MATLAB 2008. Our functions are ***clearGraphic(), preprocessing(),*** and ***postprocessing().*** The *clearGraphic()* is the base filter that use preprocessing() and postprocessing() functions and returns a black-red-white image as shown in figure 3.11. Other functions have one image parameter and contain set of steps to give an output binary image.

I2 is a thinning binary image but contains text with some unwanted data which needs more processes. So, I2 pass throw the postprocessing () and return I3. After that ***clearGraphic ()*** take I3, where I3 is a thinning black white image after removing a large amount of unwanted data. Then take I3 and apply thickens method three times to be bold as much as possible, and then convert it to RGB image to give red color for foreground and background still white. Additionally, I4 is defined to be a binary black white copy from the origin image then convert it to colored black white image. Finally, multiply I3 by I4 to get results appear in *(table 4.7)*.

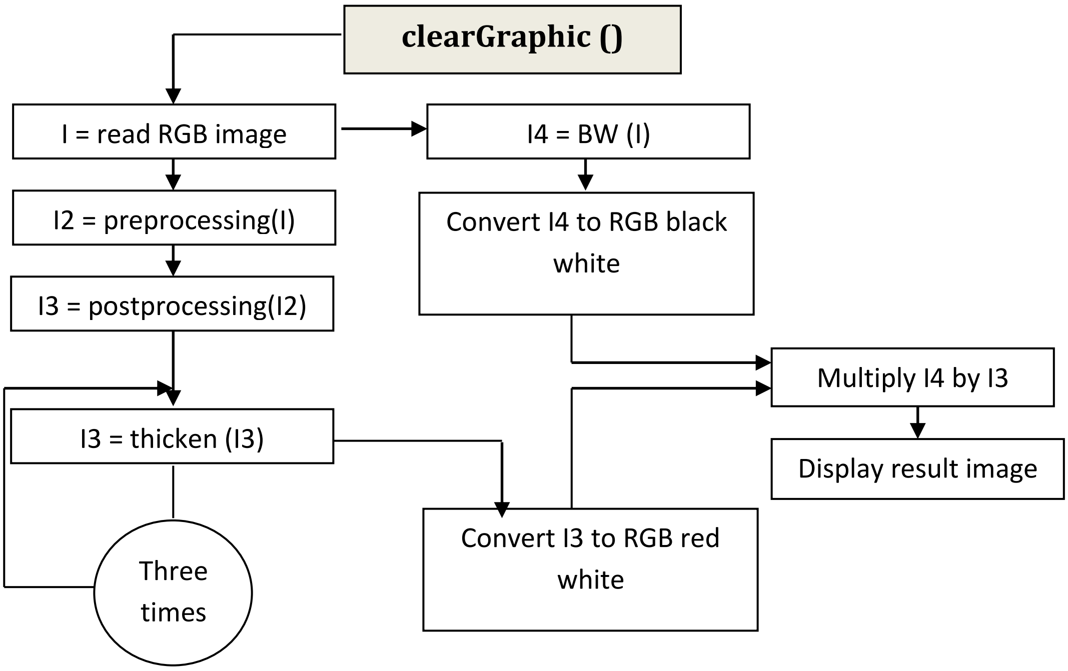


Figure 3. 11: clearGraphic () filter that represents the base structure for our work

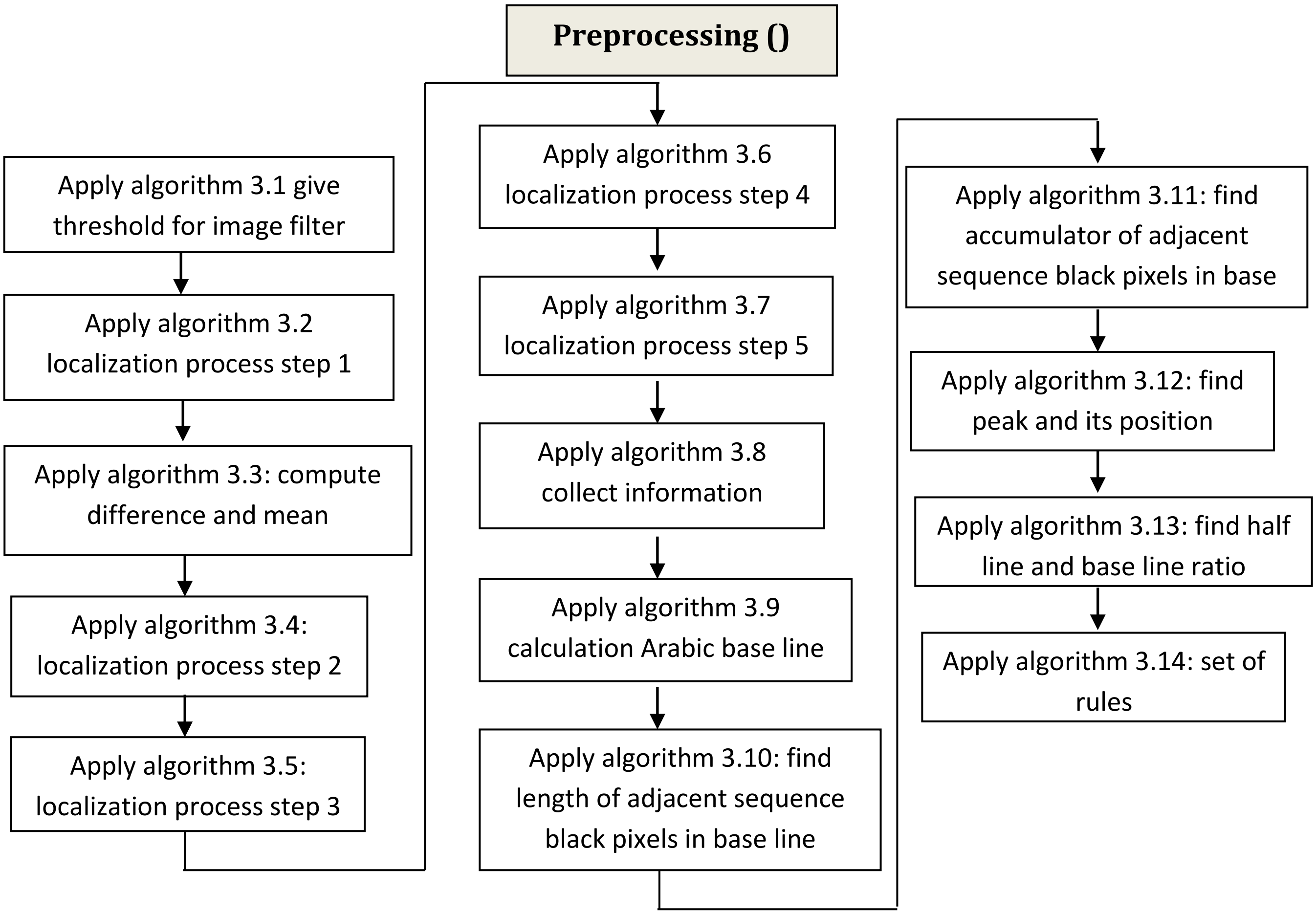

Preprocessing ()
Apply algorithm 3.1 give threshold for image filter
Apply algorithm 3.2 localization process step 1
Apply algorithm 3.3: compute difference and mean
Apply algorithm 3.4: localization process step 2
Apply algorithm 3.5: localization process step 3
Apply algorithm 3.6 localization process step 4
Apply algorithm 3.7 localization process step 5
Apply algorithm 3.8 collect information
Apply algorithm 3.9 calculation Arabic base line
Apply algorithm 3.10: find length of adjacent sequence black pixels in base line
Apply algorithm 3.11: find accumulator of adjacent sequence black pixels in base
Apply algorithm 3.12: find peak and its position
Apply algorithm 3.13: find half line and base line ratio
Apply algorithm 3.14: set of rules

Figure 3. 12: structure represents the steps of prepressing

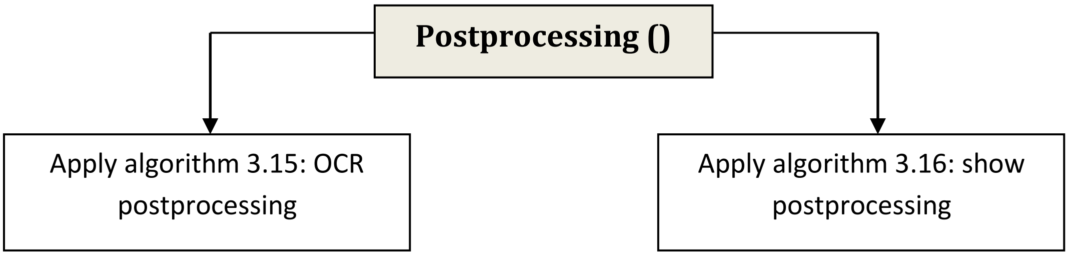

Postprocessing ()
Apply algorithm 3.15: OCR postprocessing
Apply algorithm 3.16: show postprocessing

Figure 3. 13: structure represents the two ways of postprssing

From our experience in this area, work on a features of font to find the text in the image is better than comparing each character, because it is reduce the seek time process. Thus, this algorithm could be used as a pre-step for the OCR system.

The next chapter is evaluation and discussion. For evaluation we will use two types one of them is manual and the other is automatic on set of 90 images. Finally, we will list our conclusion and future work.

Chapter Four

Evaluation and Discussion

To evaluate our work, we used two types of evaluation. First one is information retrieval context, and the second type is finding the rate of correlation between origin output and result output.

4.1: Evaluation

This section is divided into two subsections. First one is experimental settings. In this section we descript the formula of information retrieval context and also descript the correlation steps. The second is experimental results. This section describes the data set classified in groups and present result for each image on alone. Also, to presents the results for all data set in a table and for each group in another table. But I would like to note that there are two tables for each result. One of the tables is for first evaluation by using the information retrieval context. The second table is for second evaluation by using correlation method.

4.1.1: Experimental Settings

Now we will to describe the two types of evaluation techniques to represent accuracy and error rates.

The ***first evaluation*** is Micro-precision and Micro-recall formulas to evaluate our work. All measurements are applied manually and results are summarized in *table (4. 6)* and *figure (4.1).*

The ***second evaluation*** is a comparison technique. This comparison includes two partitions. The first partition is manual and the other partition is automatic. This comparison finds the rate of correlation between the origin output image and the output result.

The definition of ***Precision*** is the fraction of correct results (where ***tp*** is the true positive) to summation of correct results and unexpected results as follows:

$$\boldsymbol{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \qquad \text{Equation (3)}$$

Recall is the fraction of correct results (where ***TP*** is the true positive) to summation of correct results and missing results (where ***FN*** is false negative) as follows:

$$\boldsymbol{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \qquad \text{Equation (4)}$$

In our work Eq. (3) and Eq.(4) are used for evaluation. To measure ***Accuracy*** Eq. (5) is used, which is the fraction of correct results and correct absence (true positive and true negative) to the summation of correct results, unexpected results, missing results and correct absence (true positive ***TP***, true negative ***TN***, false positive ***FP*** and false negative ***FN*** respectively which are known as binary evaluation) .

$$\boldsymbol{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}} \qquad \text{Equation (5)}$$

$$\boldsymbol{Error} = \frac{\mathbf{FP+FN}}{\mathbf{TP+TN+FP+FN}} \qquad \text{Equation (6)}$$

"Any known measure for binary evaluation can be used here, such as accuracy, precision and recall. The calculation of these measures for all labels can be achieved using two averaging operations, called *macro-averaging* and *micro-averaging*. These operations are usually considered for averaging-precision, recall and their harmonic mean (F-measure) in information retrieval tasks. The macro-averaged and micro-averaged versions of B (TPλ, FPλ, TNλ,*FN*λ) for label λ are calculated as follows:

$B_{macro} = 1/q \ \sum_{\lambda=1}^{q} B\,(TP\lambda, TN\lambda, FP\lambda, FN\lambda)$, and $B_{micro} = B\,(\sum_{\lambda=1}^{q} TP\lambda, \sum_{\lambda=1}^{q} FP\lambda, \sum_{\lambda=1}^{q} TN\lambda, \sum_{\lambda=1}^{q} FN\lambda)$ " [30]. Therefore the additional criterion measures are Micro-Precision, Micro-Recall, Macro-Precision and Macro-Recall. So, we can explain it as follows,

Micro-average-precision is

$$\frac{\sum_{i=1}^{q} tpi}{\sum_{i=1}^{q} tpi + \sum_{i=1}^{q} fpi} \qquad \text{Equation (7)}$$

Micro-average-Recall is

$$\frac{\sum_{i=1}^{q} \mathrm{tpi}}{\sum_{i=1}^{q} \mathrm{tpi}+\sum_{i=1}^{q} \mathrm{fni}}$$ Equation (8)

Macro-average-Precision is

$$\frac{\sum_{i=1}^{q} \mathrm{Pi}}{q}$$ Equation (9)

Macro-average-Recall is.

$$\frac{\sum_{i=1}^{q} \mathrm{Ri}}{q}$$ Equation (10)

Where, ***Pi*** is the precision for each element and ***Ri*** is the recall for each element.

✓ **The evaluation steps that used in first evaluation technique:**

1. After applying text region segmentation algorithm, some words are still has alone segmentation (as word segmentation) and the most formed text region. But little words or statements do not segment from the beginning of using the connected label method. So, we count correct text and word segment region by count Arabic text region one time and other time just count the Arabic word region for each image. Then count the correct visible statements/words in output result as *correct result (TP)*.
2. We do not count the statements or words are not labeled from the beginning either Arabic or English after using the connected label method. Also, don't regard to paragraph region. Because our focus on statement region only. But if Arabic paragraph region are presenting in output result we are counting as *correct result (TP).*
3. Some items in the origins image represent in the output result, but it should not be represented like English text or shapes. So, we are counting as *unexpected output (FP)*.
4. The selected statements or pictures or shapes are correct absence from the output result we are counting as *correct absence (TN).*
5. Some items are selecting from the beginning but some rules incorrect prevent it from appearance in the output result, and we are counting as *missing (FN)*.
6. Then we are counting the correct results, missing results, unexpected results and correct absence results for each image to compute precision, recall and accuracy for each image. Then compute micro and macro precisions and recall for all images *(see table 4.9 and 4. 10*).

✓ **The evaluation steps that used in second evaluation technique:**

As mentioned this evaluation has two partitions manual and automatic:

- **Manual partition as follows:**
  1. Take a copy from each image after applying an interior filters and localization step.
  2. Save these copies in a folder
  3. Open Paint program application to erase unwanted data from the copies
- **Automatic partition as follows:**
  1. Open two images, the first one is the colored image and the other one is copied image after erase unwanted data.
  2. Count *all black pixels* in the original copied image to use it into final evaluation.
  3. Our filter call comparison function that compares each black pixel into the origin filter copy is in the same place in the output result or not.
  4. If the black pixel in the same place we computed as *correct result*.
  5. If there is a missing data or unexpected data in the output result we computed as *incorrect result*.
  6. Finally, to compute accuracy rate for each image divide correct result over all black pixels, and to compute error rate divide incorrect result over all black pixels.
  7. To compute accuracy rate for all images. Find the total of correct result for all images and divide it over the total of all black pixels for all images.
  8. To compute error rate for all images. Find the total of incorrect result for all images and divide it over the total of all black pixels for all images (see *table 4.1 and 4.2*).

4.1.2: Experimental results:

In this section we will show our work results on different levels of difficulties. These difficulties come from combination between background levels and text levels *(see table 4.1* and *4.2*). Such as image has variant background color with shapes and pictures, also includes text with any font color, size, style, and in different horizontal positions. Also, includes both Arabic and English languages. This example is the most difficult one.

Table 4. 1: variant background level complexity

Background Level	Background Color	Include Picture	Include Shape
1	One/White	Yes / No	Yes / No
2	Gradient	Yes / No	Yes / No
3	Variant	Yes / No	Yes / No

Table 4. 2: variant text level complexity

Text Level	Text Font color	Text font size (less than 75)	Text font style	Text wrap position	Include Arabic/English
1	Any	Same/ Variant	Same/ Variant	Without pictures	Arabic/English/both
3	Any	Same/ Variant	Same/ Variant	With pictures in left, right, above, below or inside.	Arabic/English/both

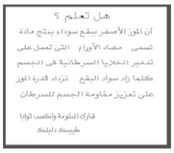
So, the rate of found of those factors with each other are affecting on the ratio of final evaluation.

Our *ClearGraphic* filter is applied on about 90 images. The types of these images are JPEG, PNG, and BMP. Each image that has own nature and there is no similarity between of them. So, we are classifying them into four groups (A, B, C, and D), (see table *C.1*, *C.2, C.3, and C.4*), respectively. Results are depicted in *(table 4.3 and C.5).*

Group A contains images with variance text written on white background. ***Group B*** contains images with variance text written on background filled with one color rather than white. ***Group C*** contains images with variance text written on background filled with gradient/variant color. ***Group D*** contains images with variance text written on picture background.

Sample of final output results that should be given to OCR system are shown in the next table:

Table 4. 3: sample results for OCR system

1. 	2. 	3. 	4.
5. 	6. 	7. 	8.
9. 	10. 	11. 	12.

After displaying the data set group and result for each image. We put now the aggregated results for all 90 images and for each group separately. We use four kinds of data representation to represent the results. First one is Table shape. The second is Doughnut chart. Third is Pie chart. Last one is Column chart.

Table 4. 4: results for all above 90 images and first of 50 images using first evaluation.

First evaluation	Macro-precision	Macro-recall	Macro-accuracy	Error	Micro-precision	Micro-recall
90 image	86.08%	76.14%	90.53%	8.86%	86.96%	87.23%
50 image	87.94%	92.08%	92.26%	9.58%	86.90%	89.46%

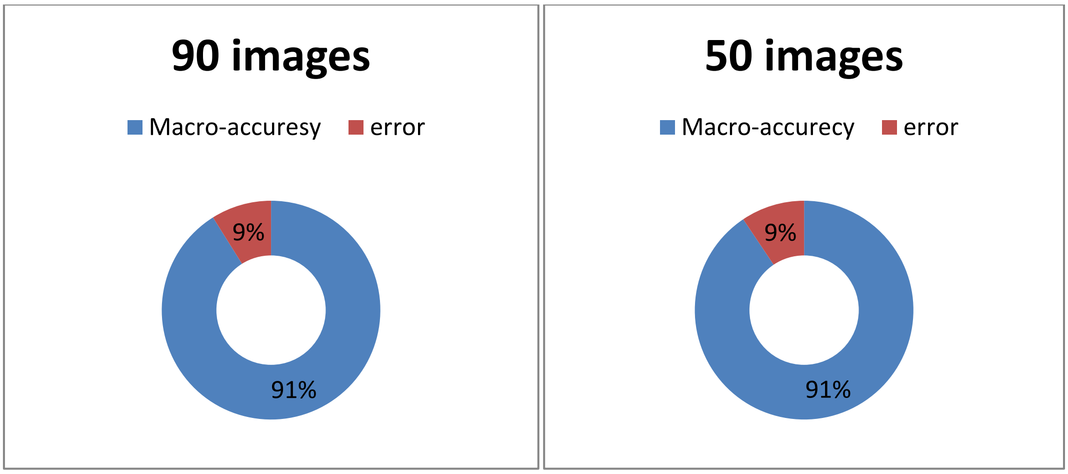


Figure 4. 1: comparison between 50 and 90 image according accuracy and error ratio using first evaluation

Table 4. 5: represents results for each group A, B, C, D, and E using first evaluation

First evaluation	Group A (19)	Group B (15)	Group C (22)	Group D (34)
Macro-precision	84.90%	91.60%	87.35%	83.48%
Macro-recall	91.34%	94.28%	83.75%	80.95%
Macro-accuracy	90.86%	93.93%	88.46%	90.18%
Error	8.06%	3.87%	10.13%	9.77%
Micro-precision	86.57%	93.94%	89.15%	81.23%
Micro-recall	91.22%	96.88%	88.63%	78.40%

Table 4. 6: represents values for each 50 and 90 item using second evaluation

Second evaluation	**All black pixel**	**All black in**	**All black out**	**Accuracy**	**Error ratio**
90 image	297346	271312	26109	91.24%	8.78%
50 image	207712	190264	17511	91.59%	8.43%

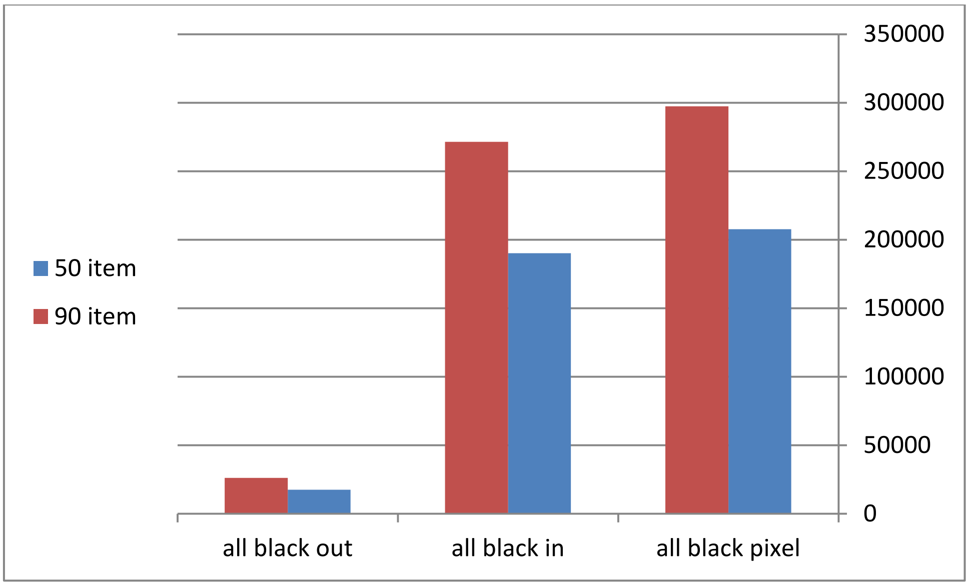


Figure 4. 2: comparison between 50 and 90 image according number of black pixels all, in and out using second evaluation

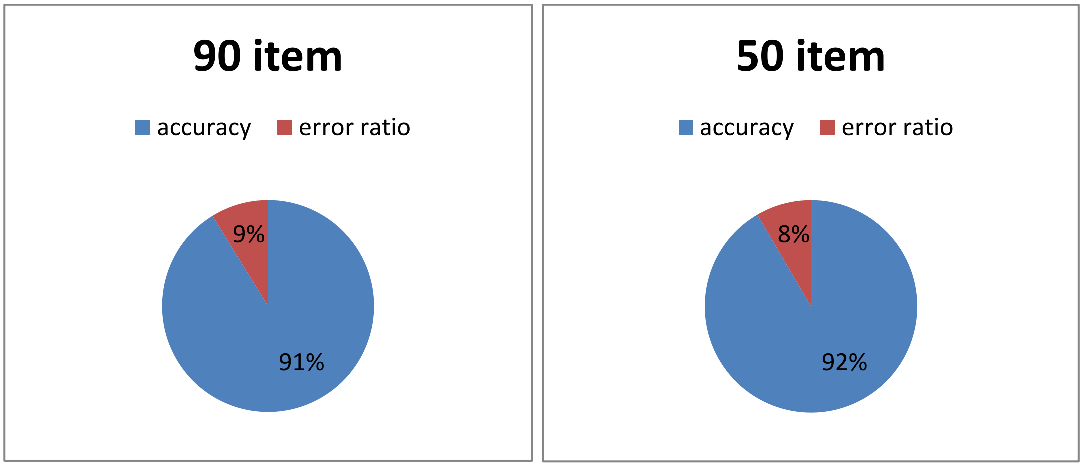


Figure 4. 3: comparison between 50 and 90 image according accuracy and error ratio using second evaluation

Table 4. 7: represent values for each group using second evaluation

Second evaluation	Group A (19) image	Group B (15) image	Group C (22) image	Group D (34) image
Total black pixels in the origin image after applying filter	65195	43730	123738	64683
Black in (correct position)	60374	42569	110192	58177
Black out (missing and unexpected)	4821	1170	13612	6506
Accuracy	92.61%	97.35%	89.05%	89.94%
Error	7%	3%	11%	10%

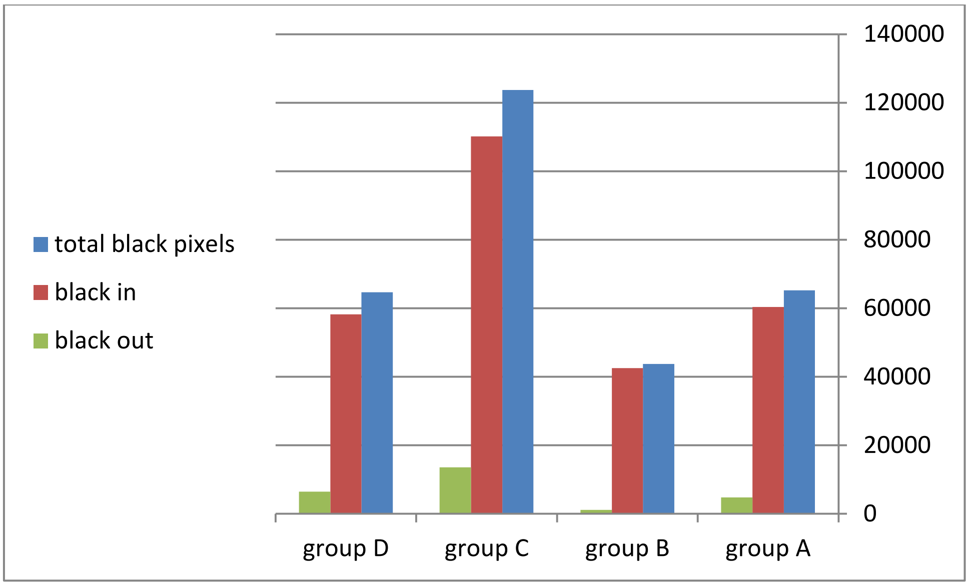


Figure 4. 4: comparison between groups A, B, C, and D according number of black pixels all, in and out using second evaluation

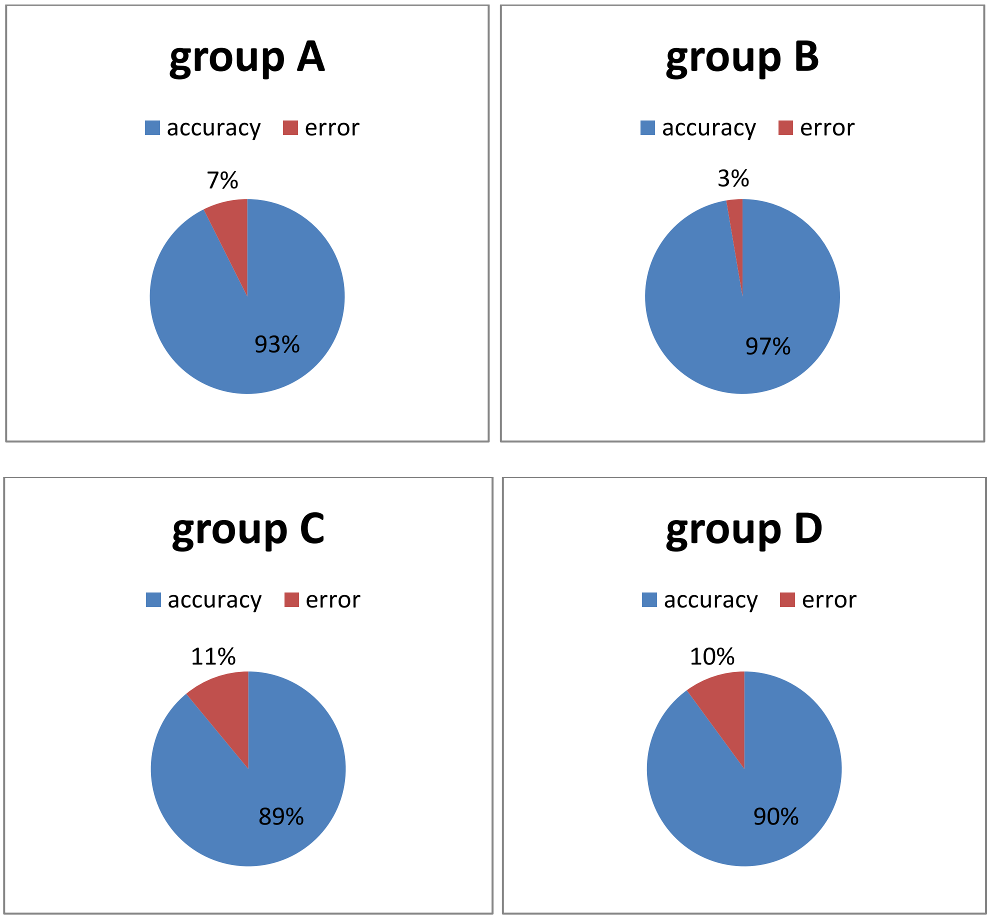


Figure 4. 5: comparison between groups A, B, C, and D according accuracy and error ratio using second evaluation

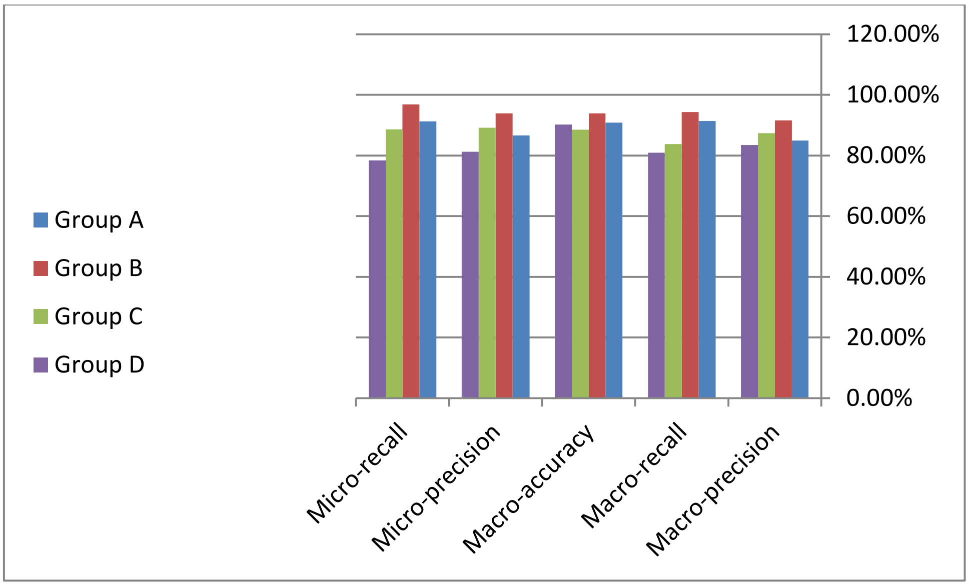


Figure 4. 6: comparison between groups A, B, C, and D according accuracy and error ratio using first evaluation

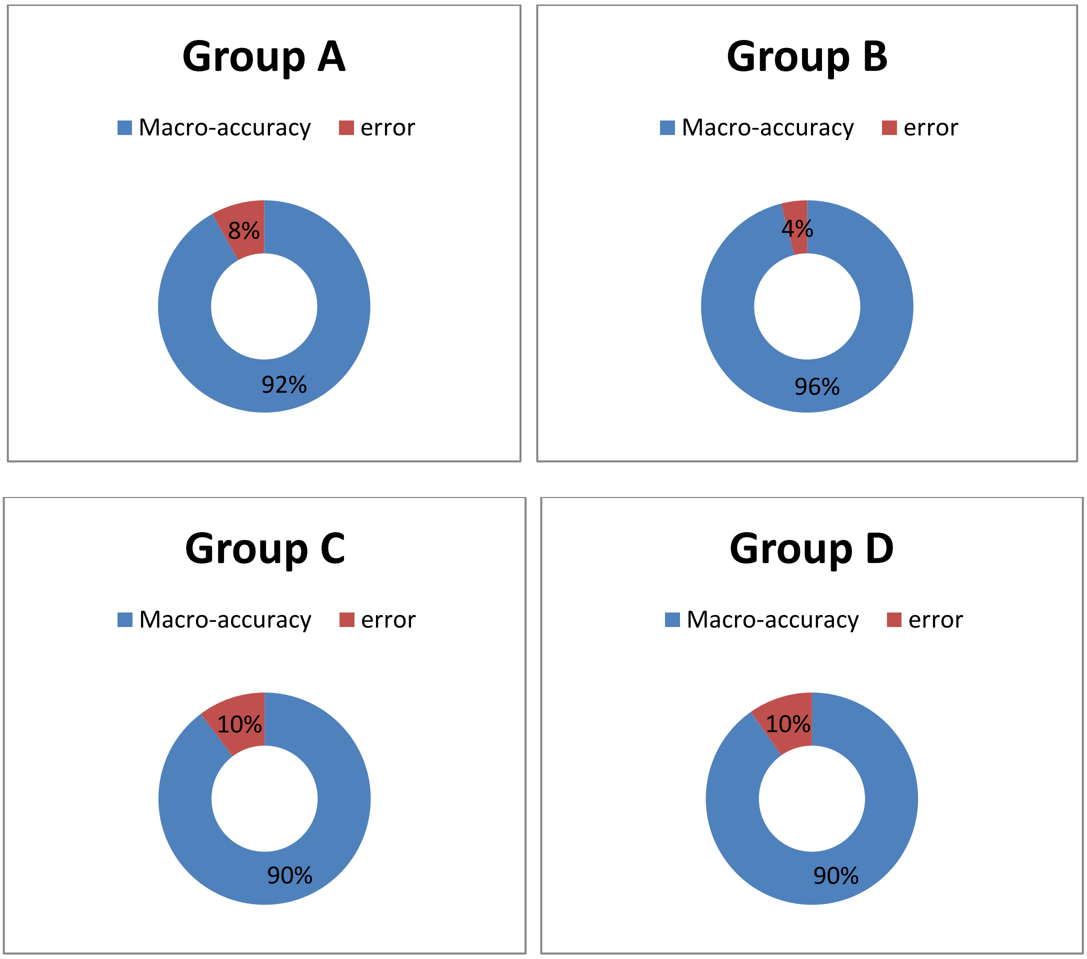


Figure 4. 7: comparison between groups A, B, C, and D according accuracy and error ratio using first evaluation

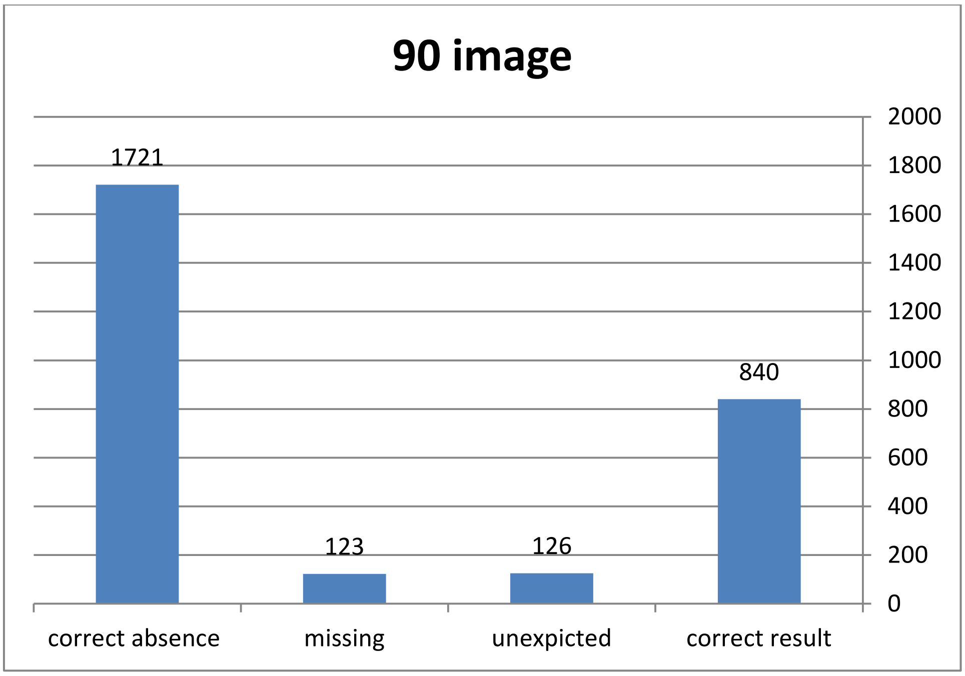


Figure 4. 8: compare between values of correct result, correct absence, unexpected, and missing results for all 90 images.

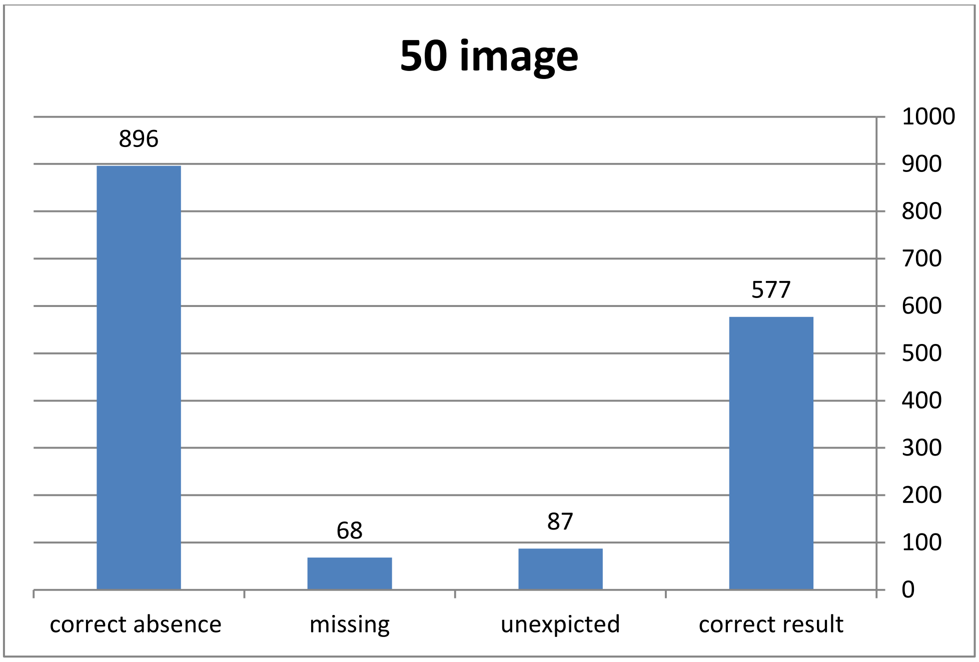


Figure 4. 9: compare between values of correct result, correct absence, unexpected, and missing results for first of 50 images.

We experiment our work on ninety images. These images are collected from web pages like Google search and Facebook in random way. We are select images have a variant size font.

4.2: Discussion

We have some observations on the algorithm works and its results especially on micro-recall, micro-precision, correct result, missing, unexpected and correct absence values.

Our observations are representing as follows:

1. From *(figure 4.2 and 4.4*) you can see that by *second evaluation*, the amount of black pixels is should equal to the actual black pixels that should be represents in the output result are very close. And the rate of missing and unexpected black pixels is very small. Also, from *(figure 4.3 and 4.5)* you can see that by *second evaluation*, accuracy is around 90% and the rate of error is under 11%. This means our proposed solution is deal will and can discover Arabic base text in a proper way.
2. From (table 4.5) we can see the maximum accuracy is in group B, because if you look to the pictures in group B you can observe that the font style in each image is mostly the same. This means that, if the image contains Arabic text with the same font type and font size the success ratio will be increase.

3. The ClearGraphic filter gives high accuracy if the complex image includes text with clear distance between lines so it does not have word close, and the words in the same line have the same style. Thus, the ClearGraphic filter is working correctly on a single statement, and sometimes on a single word.

4. The ClearGraphic filter works on different style of images, therefore, the text should be clear and the interior filters could discover the text.

5. Has been deleted group of images because the actual output should be a clear image without any black pixels, so the accuracy and error for it is NAN.

6. Sometimes the correct output could be 50% or less. And unexpected output more than 50%ffor more than one reason. First one is the image include different font size specially 12 or less with 40 or more. So the distance factor in ClearGraphic filter should be affected and the font with small size only shows response. Another reason after applying interior filter there are some connection between lines. This connection makes two lines as one line and in this case we cannot detect if it is a text or not.

7. We found ClearGraphic filter can detect some text wrote in diagonal rotation, because the rotation angle is not large. But does not work on text write by word art, because after using the interior filters the words have outline border without filling the inside, for this reason our algorithm couldn't detect it.

8. If we make our conditions more restriction we give a results are absolutely Arabic with little uncorrected results. But the restricted conditions effect on the ratio of retrieved Arabic text. Therefore we make our conditions have little flexibility to retrieve the highest amount of Arabic texts with some uncorrected and absence results was observed.

9. Our algorithm works correctly on a small size font. The font size is approximately from eight to thirty. Also, works on bigger than thirty but maybe lost some of them. We think that back to the font type if it is normal font with simple format or it has word art format. We look that if the text has word art format the ratio of missing is very big.

10. From experiment, if the image contains only Arabic text without any English text and has clear background color. We mean in clear background, the background has one color not picture and there is no part of pictures near to the text then the segmentation process works

successfully. Because the value of distance is consider as a factor in the segmentation process. If all factors are available then recall, precision and accuracy are equal to one.

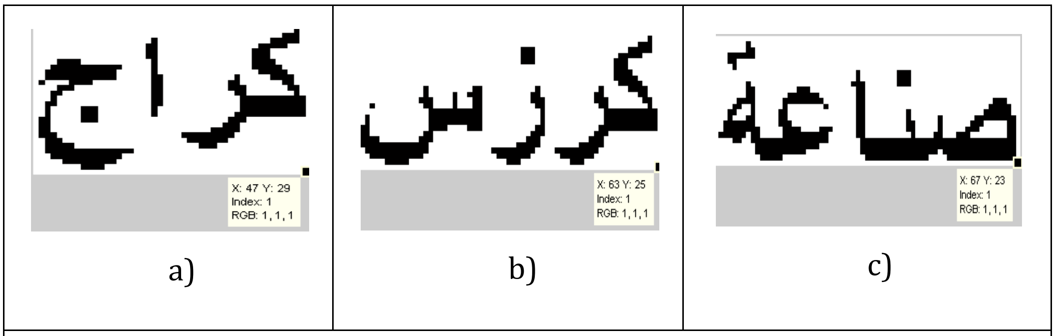
11. From our experience in this area. If we relied on our work on edge map the segmentation process works successfully one hundred percent. Because the high intensity gives to the text and other parts of the image has low intensity. So the edge map is perfect solution for complex background and to extract most amount of text automatically from complex image.

12. From *(figure 4.8 and 4.9) by first evaluation.* You can see the amount of unwanted item in each chart is very high. This means our algorithm it gives high accuracy. And it is work will to identify unwanted items and prevent it to display into final output result. In addition the amount of missing and unexpected items is very low.

13. If the image has text region rather than word/character region (i.e.: long string line) this gives a strong factor to success appear Arabic text. For example, in large font size the character "Alef" may has character segmentation. So, when apply thinning method we take a thin vertical line that may to be a garbage line because it does not has base line. Thus for that the character was removed.

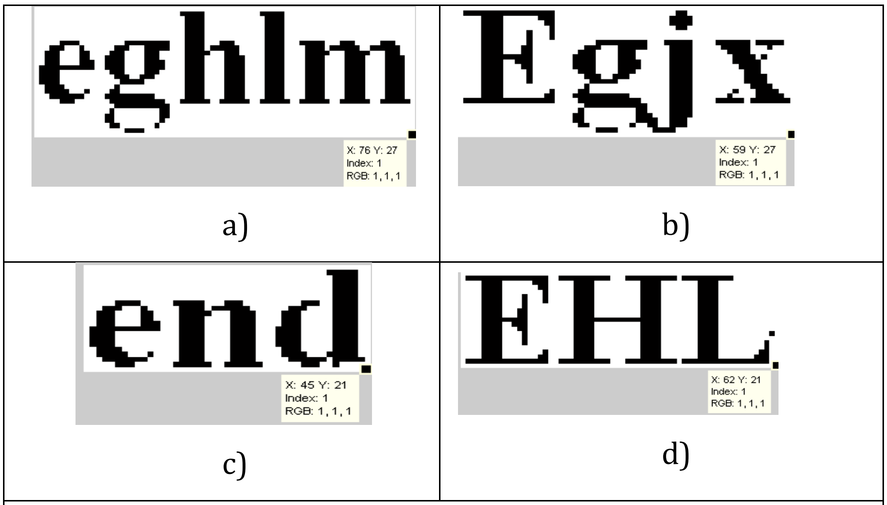
14. From our experiments in this area the reason that displays some English text on the final results. We observed that for example, the English font with size twenty two and font-style is Times New Roman has different heights approximately between twenty one and twenty seven as you see in (Figure 4.10). But the heights in Arabic with the same format are between twenty three and twenty nine as you see in (Figure 4.11). So, there is some interlacing between of them in the conditions. It is difficult to separate Arabic from English. So, we think that there is a sensitive

factor could be used to remove the interlacing between Arabic and English at all. This sensitive factor needs more image analysis.



a) b) c)

Figure 4. 10: Arabic text with parts of characters wrote in above and below the base line. a) Has characters Geem gives long height. b) All characters under the base line have the same height. c) All characters wrote on the base line.



a) b) c) d)

Figure 4. 11: English text with parts of characters wrote in above and below the base line. a) Has small characters with G gives long height. b) Capital and small characters have the same height of (a). c) and d) All characters wrote on three lines have the same height.

15. We test our algorithm on Scand image like personal card. We are not sure it gives a result. But the surprise is our algorithm gives acceptable results. And make another test on image that takes via print screen button from desktop and web site. And also works well too.

Finally, from our research exactly on how could be evaluate our work. We see that each work evaluate his work in variance ways depending on his job. Some of them use precision and recall rate. Other people use detection rate and accuracy. And some other can't be evaluated because the subject is very difficult. So, we use precision and recall to compare our work result with other works. And make correlation evaluation to make sure our result. And we see the correlation rate is suitable evaluation in this subject area because may contain character, word or string region which is make the rate absence of character equal the rate absence of string.

The experimental results for precision and recall are compared to work in [21], where their precision is 72.8% and their recall is 78.54% for 50 samples. But in our work precision for 50 images is 86.90% and recall is 89.46%. But we need to say here, in [21] Works to extract Farsi text from video not from static image. Our algorithm tries to extract the text from photo image, and its work well. As well as for different font type and size.

Chapter Five

Conclusion and Future work

This chapter is the last one that presents our conclusion as a summary for all our work. Then present our future work.

5.1: Conclusion

In this section we conclude our research by outlying our proposed method and summary its primary performance indicator and point out its limitation.

Our approach is applied on colored/complex image that contains horizontal Arabic and English text to extract Arabic text. Our approach summarized in three steps. 1) Pre-processing, 2) Text detection, and 3) Post-processing. Use Wiener2 and rangefilt filters and automatic thresholding in pre-processing step. Also prepare an output result with white background and red marking color for text detection, also a copy for OCR system using set of rules. These rules depending on 1) compute the ratio of item area over segment area 2) identify base line and 3) find maximum peak. In post processing step, we enhance the results by removing unwanted items from the results come from text detection step.

We evaluate the proposed approach on a comprehensive dataset including variety of image samples gathered by our self. Using 90 samples and evaluated by two types of evaluation. The first evaluation by information extraction retrieval is overall precision 86.96%, recall 87.23%, and an error rate is 8.86%. The second evaluation by correlation is overall accuracy 91.24% and error ratio 8.78% is obtained.

The ClearGraphic filter is applied on different style of images with variance of difficulties, but there are some limitations as follows: 1) ClearGraphic filter does not work on diagonal orientations. 2) Does not distinguish between English and Arabic in a proper way. 3) There is still little noise in final results that needs enhancement. 4) ClearGraphic filter could be localize all text but after applying text type detection method there are some Arabic character/word or text are avoid it.

5.2: Future Work

- After finish the segmentation process enter the segmented parts into machine learning with extracting feature to make a text classification method which minimize error ratio.

- Extract diagonal Arabic text. We have some attempt, depending on computing the diagonal direction and its angle then make a rotation to that angle in clock wise minus 10. We are depending on our work on a ready method done by Gonzalez. It works successfully but needs more preprocessing steps to be ready to work in our algorithm.

- Enhance our results by create a method that can detect South West angle. This angle helps us to detect Arabic text in easy way in a paragraph region.

- Enhance segmentation process, such as find a way to display texts that remove by rangefilt like using inverse method. Also, use filter to remove noise from final output.

- Add our ClearGraphic filter to MATLAB library as an enhancement filter in image processing.

- Integrate ClearGraphic filter with OCR system to increase the chance of extracting text from complex image.

Bibliography

[1] Abuhaiba, I. S. (2006). Segmentation of Discrete Arabic Script Document Images. *Journal of Al Azhar University–Gaza (Natural Sciences) , Vol. 8, ISSN 1810-6366*, 85-108.

[2] Ahmed M. E., Mohamed A. I. (2000). A Graph-Based Segmentation and Feature Extraction Framework for Arabic Text Recognition.

[3] AIM, (2001). Optical Character Recognition. Technical paper, *AIM,Inc. 634 Alpha Drive, 2-10.*

[4] Amin, A., S. Fischer, T. Parkinson, and R. Shiu. (1996). Fast Algorithm for Skew Detection. *IS&T/SPIE, symp. On Elec. Image, San Jose,* USA.: 65 – 76.

[5] Anthony C., Mohammed B., Neil B. (2001). An Arabic optical character recognition system using Recognition Based Segmentation. *Pattern Recognition 34* , 215-233.

[6] El-Mahallawy, M. S. (2008). *A LARGE SCALE HMM-BASED OMNI FONT-WRITTEN OCR SYSTEM FOR CURSIVE SCRIPTS.* Giza, Egypt: Faculty of Engineering, Cairo University.

[7] Jain A. K., Yu B. (1998). Automatic Text Location in images and video Frames. Pattern Recognition, Vol. 31, No. 12: 2055-2076.

[8] Kareem D. and Ossama E. (2007). Retrieving Arabic Printed Document: a Survey.

[9] KASMIRAN J. and MOHAMED A. (2002). A SURVEY AND COMPARATIVE EVALUATION OF SELECTED OFF-LINE ARABIC HANDWRITTEN CHARACTER RECOGNITION SYSTEMS. *Jurnal Teknologi, 36(E) , Jurnal Teknologi, 36(E) Jun 2002: 1–18*, 1–18.

[10] Keechul J., Kwang I. K., Anil K. J. (2004). Text Information Extraction in Images and Video: A Survey. 1-35

[11] Kim K. C., Byun H. R., Song Y. J., Choi Y. W., Chi S. Y., Kim K. K., Chung Y. K. (2004). Scene Text Extraction in Natural Scene Images using Hierarchical Feature Combining and Verification. IEEE.

[12] Li C., Ding X. Wu Y. (2001). Automatic Text Location in Natural Scene Images. Proc. of 6th ICDAR: 1069-1073.

[13] Magazine, M. E. (2007). *Essay On Arabic Ocr Packages In The Market_Windows .* Retrieved from www.itp.net/Arabic.

[14] Maher K. and Abdelfettah B. (2009). Towards A Distributed Arabic OCR Based on the DTW Algorithm: Performance Analysis. *The International Arab Journal of Information Technology , Vol. 6, No. 2* .

[15] Mahmoud, S. (1994). Arabic Character Recognition using Fourier descriptors and character contour encoding Pattern Recognition. 27(6); 815-824.

[16] Matas J. Chum O., Urban M., Pqjdlo T. (2002) . Robust Wide base line Stereo form maximally stable extremal regions. In proceeding of British Machine Vision Conference (BMVC): 384-393.

[17] Michael Bukland and Fredric Gey. (1994). The Relationship between Recall and Precision. . *Journal of the American society for information science* , 12-19 .

[18] Michael D., Horst B. , Silk W. (2008). Using Web Search Engines to Improve Text Recognition. IEEE.

[19] Michael R. Lyu, F. I. (FEBRUARY 2005). A Comprehensive Method for Multilingual Video Text etection, Localization, and Extraction. *VOL. 15, NO. 2*, 243-255.

[20] Mikolajczyk K., Tuylelaars T., Schmid C., Zisserman A., Matas J., Schaffalitizky F., Kadir J. and Van G. L. (2005). A comparison of a affine region detectors. International Journal of Computer Vision (IJCV), 65 (1-2): 43-72.

[21] Mohieddin Moradi, Saeed Mozaffari, and Ali Asghar Orouji. (2010). Farsi/Arabic Text Extraction from Video Images by Corner Detection. *IEEE* .

[22] Nazif, A. (1975). A System for the Recognition of the Printed Arabic Characters. Master thesis, Faculty of Engineering, Cairo University.

[23] Ohya J., Shio A., Akamatsu S. (1995), Recognizing Characters in Scene images, IEEE.

[24] Omar Al-J., Samer Al-K., Bashar Al-G., Mohamed F., Hani K. (2000). A New Algorithm for Arabic Optical Character Recognition. 1-14.

[25] Paul H., Ben H., Chris J., Linda V. G., Amlan K. (1997). Optimizing OCR accuracy for bi-tonal, noisy scans of degraded. *The MITRE Corporation, 7515 Colshire Drive, McLean, VA 22102-7508. , Proc. of SPIE Vol. 5817*, 179-187.

[26] Prof. Mohsen A. A. Rashwan and Dr. Mohamed Attia. (2008). Academic history of Arabic OCR of RDI, and principal/co-principal researchers;.

[27] RASHWAN M.A.A., FAKHR M.W.T., ATTIA M. , EL-MAHALLAWY M.S.M. (2007). *ARABIC OCR SYSTEM ANALOGOUS TO HMM-BASED ASR SYSTEMS; IMPLEMENTATION AND EVALUATION.*

[28] Tarek A. E., Aly A. F. (2009). English/Arabic Cross Language Information Retrieval (CLIR) for Arabic OCR-Degraded Text. *Communications of the IBIMA , Volume 9, ISSN: 1943-7765*, 208-218.

[29] Tolba, M., S. Wahab, and A. Salem. (1987). A Recognition Algorithm for Printed Arabic Character. Proc. IASTED inter. Symp. In applied information, Switzerland, 128-131.

[30] Tsoumakas, G., Katakis, I., & Vlahavas, I. P. (2010). Mining multi-label data. In O. Maimon, & L. Rokach (Eds.) . In *Data Mining and Knowledge Dis- covery Handbook* (pp. 667-685). , Heidelberg, Germany: Springer-Verlag, 2nd ed.

[31] UP, o. e. (2009). chapter 8 Evaluation in information retrival.

[32] Volker M., Haikal El. A. (2009). Arabic Word and Text Recognition -- Current Developments. *Technische Universitaet Braunschweig Institute for Communications Technology (IfN) Schleinitzstrasse 22, 38106 Braunschweig Germany* , 31-36.

[33] What Resolution Should Your Images Be?

[34] Yao-Yi C., Craig A. K. (2011). Recognition of Multi- oriented, Multi-Sized and Curved Text.

[35] Zhidong L., Issam B., Andras K., John M., Premkumar N., and Richard S. (n.d.). A Robust, Language-Independent OCR System.

[36] Zied T., Mohamed L. and Maher K. (2011). Arabic Cursive Characters Distributed Recognition using the DTW Algorithm on BOINC: Performance Analysis. *(IJACSA) International Journal of Advanced Computer Science and Applications , Vol. 2, No.3*, 75-79.

[37] Zhong Y., Shang and Jain A. K. (2000). Automatic Caption localization in Compressed Video. IEEE Trans. on PAMI, Vol. 22, No. 4: 385-392.

[38] Zhong Y., Karu K., Jain A. K. (1995). Locating Text in Complex Images. Pattern Recognition. Vol. 28, No. 10: 1523-1535.

[39] Character Recognition by Feature Point Extraction, http://www.ccs.neu.edu/home/feneric/charrec.html , 2/5/2012

[40] Free OCR, http://www.free-ocr.com/, 2/5/2012.

[41] Free Online OCR Convert JPEG, PNG, GIF, BMP, TIFF, PDF, DjVu to Text, http://www.newocr.com/, 2/5/2012.

[42] GOCR, http://jocr.sourceforge.net , 2/5/2012.

[43] Google Operating System, Open-Source OCR Software, Sponsored by Google http://googlesystem.blogspot.com/2007/04/open-source-ocr-software-sponsored-by.html, 2/5/2012.

[44] lifehacher, five best text recognition tools, http://lifehacker.com/5624781/five-best-text-recognition-tools, 2/5/2012.

[45] MakeUseOf, http://www.makeuseof.com/tag/top-5-free-ocr-software-tools-to-convert-your-images-into-text-nb/, 2/5/2012.

[46] RDI, OCR Arabic omni font-written OCR, http://www.rdi-eg.com/projects/OCR.htm, 20/3/2012.

[47] SimpleSoftware, http://www.simpleocr.com/OCR_Software_Guide.asp, 5/4/2012.

[48] tjansson.dk, OCRopus – open source text recognition, http://www.tjansson.dk/?p=1616, 2/5/2012.

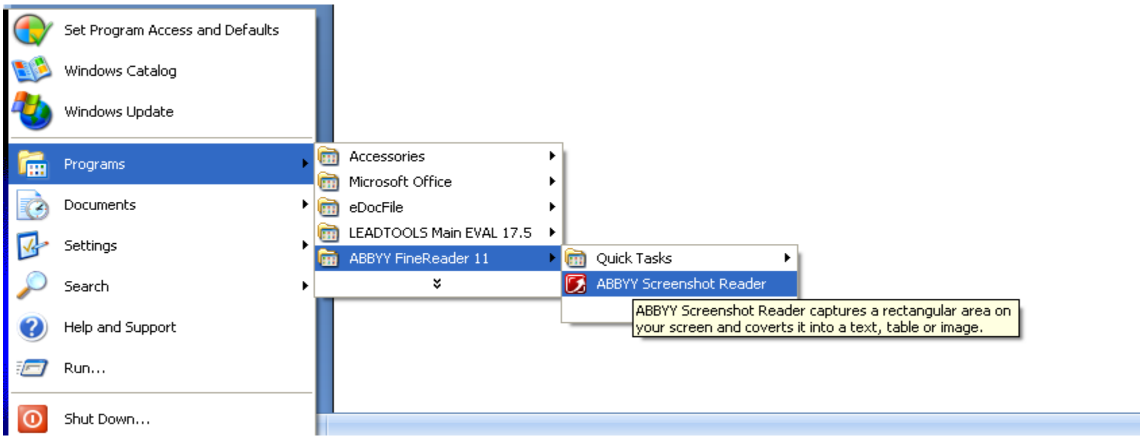
[49] Webopedia, optical character recognition, http://www.webopedia.com/TERM/O/optical_character_recognition.html, 2/5/2012.

[50] Yshowtopapp, http://www.yshow.net/show-all-all-all-MobiReader+Biz%2B++-+Business+Card+OCR+Reader+ (Korean+%26amp%3B+English)-0-368180313.html, 5/4/2012.

Appendix A

Details about ABBYY Finereader 11

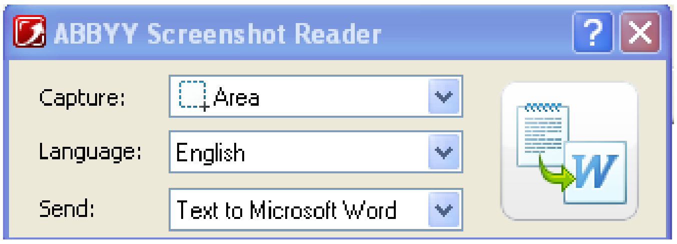
Exclusive details about ABBYY FineReader 11 are given here. How to use it and what are the results if we try to test Arabic graphics image?



ABBYY Screenshot Reader captures a rectangular area on your screen and coverts it into a text, table or image.

Figure A. 1: to run ABBYY Screenshot Reader

After install ABBYY FineReader 11, you found it in programs window. And to try it you can choose ABBYY ScreenShot Reader or Quick Tasks depends on the kind of use you need. This program you can apply it by take a screenshot which like as print-screen function to get an image from any part you need on your screen as you see in *(figure A.3 and* A.5), or you can browse the image place and select the image file as you see in (*figure A.10, A.14 and A.16)*. At first, we select an ABBYY ScreenShot Reader to apply it on two web sites, one of them is English and the other is Arabic to see if it is work or not and how will the results form. To run this program *see (figure A*.1).



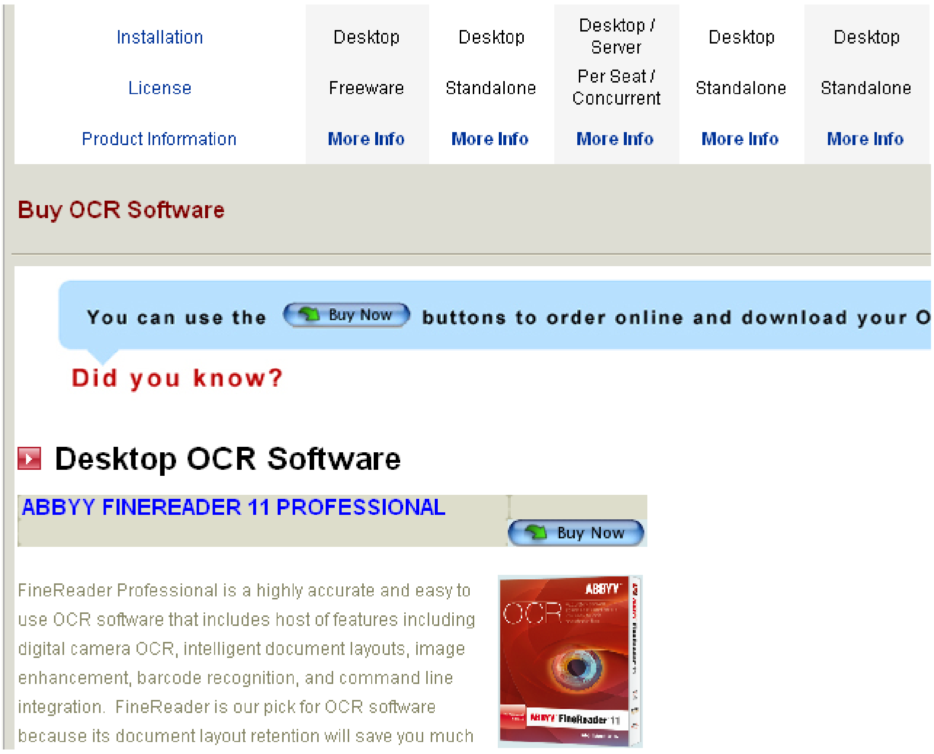
Capture: Area

Language: English

Send: Text to Microsoft Word

Figure A. 2: The ABBYY Screenshot Reader window

In (*figure A.2)*, you must choose Capture to identify the shape of selection area. That may take (Area, window, screen, Timed Screen). Also choose the Language to be used as a data set to compare the text in the image. The common languages used are (English, German, French, Spanish and Italian); also there is another choice to select more Languages. Finally, select from Send list-box the type of the document you want to convert to it. There are many types like (text, word, table and image). Then click on the button beside the list-boxes. Then you can see the capture you are select from ABBYY Screenshot Reader. To try it on English page, choose your preferred part and then click Enter from key-board. The part you select convert to image as you see in (*figure A.3)*. Then after that, automatically give a result as a text on a word document as you selected, see *(figure A.4)*. It gives an excellent result for extracting English text with little mistakes to identify character D. We tried the same program on Arabic web site as we see in *fig 5*. The result represents a garbage characters as we see in (*figure A.6*), because Arabic choice is not in the language list-box, see(*figure A.7)*.



Installation	Desktop	Desktop	Desktop / Server	Desktop	Desktop
License	Freeware	Standalone	Per Seat / Concurrent	Standalone	Standalone
Product Information	More Info	More Info	More Info	More Info	More Info

Buy OCR Software

You can use the Buy Now buttons to order online and download your O

Did you know?

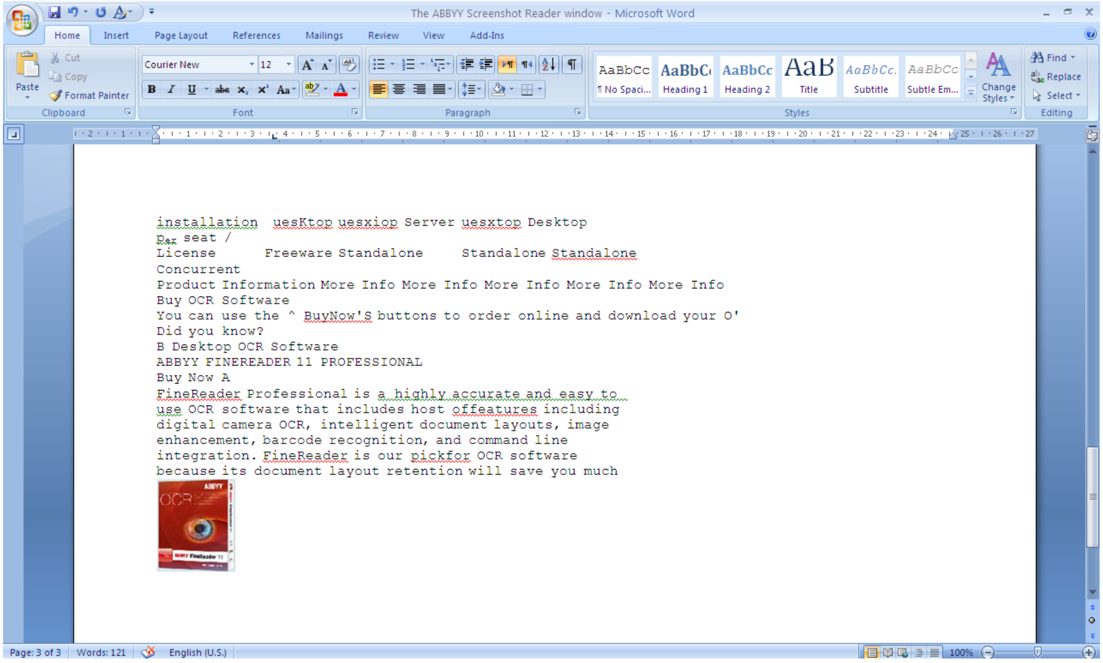
Desktop OCR Software

ABBYY FINEREADER 11 PROFESSIONAL

Buy Now

FineReader Professional is a highly accurate and easy to use OCR software that includes host of features including digital camera OCR, intelligent document layouts, image enhancement, barcode recognition, and command line integration. FineReader is our pick for OCR software because its document layout retention will save you much

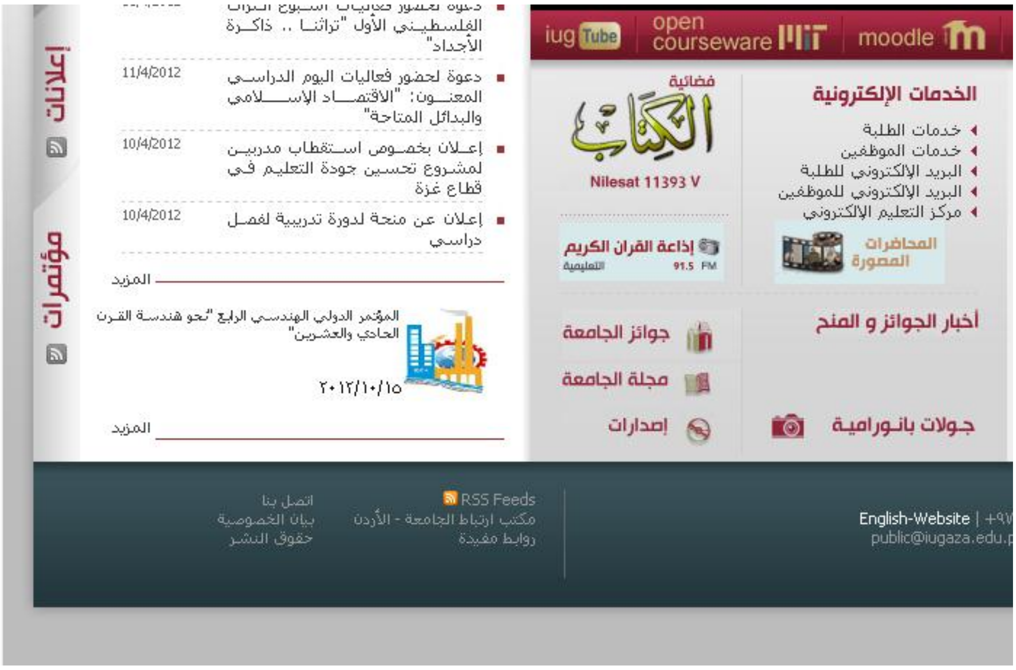
Figure A. 3: Example takes by ABBYY Screenshot Reader by click Enter key for English page



```
installation  uesKtop uesxiop Server uesxtop Desktop
Der seat /
License     Freeware Standalone     Standalone Standalone
Concurrent
Product Information More Info More Info More Info More Info More Info
Buy OCR Software
You can use the ^ BuyNow'S buttons to order online and download your O'
Did you know?
B Desktop OCR Software
ABBYY FINEREADER 11 PROFESSIONAL
Buy Now A
FineReader Professional is a highly accurate and easy to
use OCR software that includes host offeatures including
digital camera OCR, intelligent document layouts, image
enhancement, barcode recognition, and command line
integration. FineReader is our pickfor OCR software
because its document layout retention will save you much
```

Figure A. 4: the result of extracting the text in image was appearing in figure A.1.

:



دعوة لحضور فعاليات اسبوع التراث الفلسطيني الأول "تراثنا .. ذاكرة الأجداد"

11/4/2012 دعوة لحضور فعاليات اليوم الدراسي المعنون: "الاقتصاد الإسلامي والبدائل المتاحة"

10/4/2012 إعلان بخصوص استقطاب مدربين لمشروع تحسين جودة التعليم في قطاع غزة

10/4/2012 إعلان عن منحة لدورة تدريبية لفصل دراسي

المزيد

المؤتمر الدولي الهندسي الرابع "نحو هندسة القرن الحادي والعشرين"

٢٠١٢/١٠/١٥

المزيد

الخدمات الإلكترونية
- خدمات الطلبة
- خدمات الموظفين
- البريد الإلكتروني للطلبة
- البريد الإلكتروني للموظفين
- مركز التعليم الإلكتروني

أخبار الجوائز و المنح

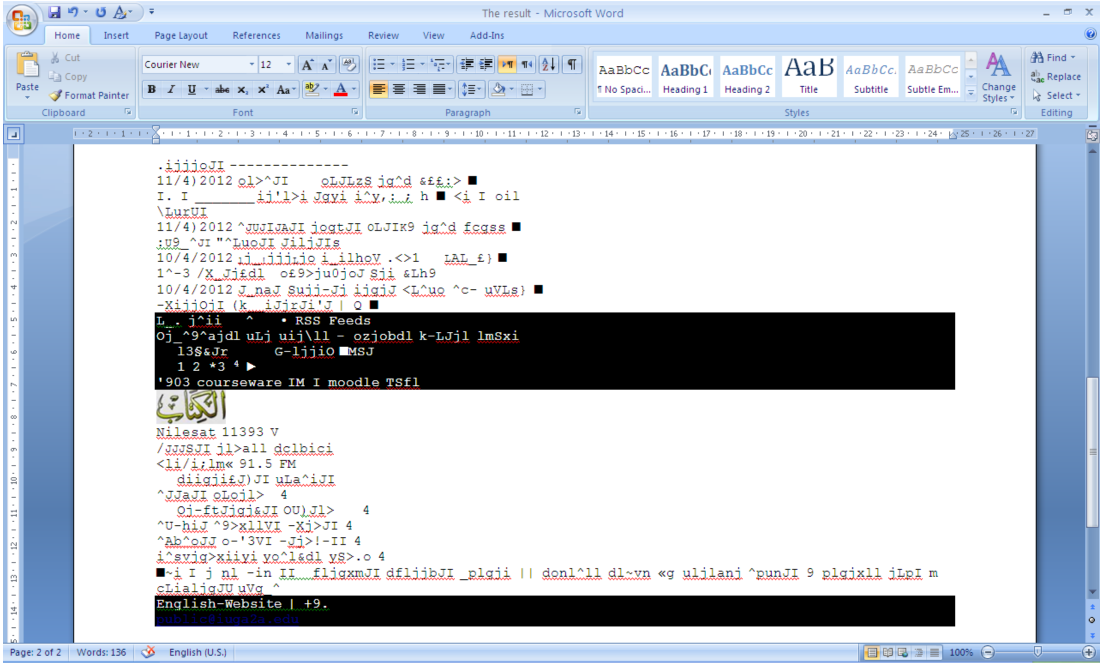
جولات بانورامية

مكتب ارتباط الجامعة - الأردن
روابط مفيدة

اتصل بنا
بيان الخصوصية
حقوق النشر

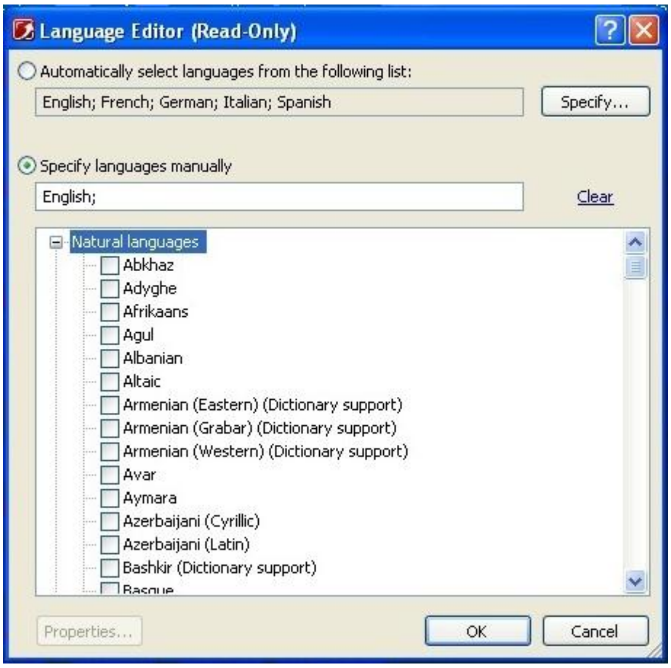
English-Website | +٩٧
public@iugaza.edu.p

Figure A. 5: Example takes by ABBYY Screenshot Reader by click Enter key for Arabic page



```
.ijjjoJI --------------
11/4)2012 ol>^JI    oLJLzS jg^d &££:> ■
I. I ________ij'l>i Jgvi i^y,:_; h ■ <i I oil
\LurUI
11/4)2012 ^JUJIJAJI jogtJI oLJIK9 jg^d fcgss ■
:U9_^JI "^LuoJI JiljJIs
10/4/2012 ;j_;jjj;jo i_ilhoV .<>1   LAL_£} ■
1^-3 /X_Jj£dl  o£9>ju0joJ Sji &Lh9
10/4/2012 J_naJ Sujj-Jj ijgjJ <L^uo ^c- uVLs} ■
-XijjOjI (k__iJjrJi'J | Q ■
L_. j^ii    ^    • RSS Feeds
Oj_^9^ajdl uLj uij\ll - ozjobdl k-LJjl lmSxi
   13§&Jr      G-ljjiO ■MSJ
   1 2 *3 4 ▶
'903 courseware IM I moodle TSfl
Nilesat 11393 V
/JJJSJI jl>all dclbici
<li/i;lm« 91.5 FM
   diigjijJ)JI uLa^iJI
^JJaJI oLojl>  4
   Oj-ftJjgj&JI OU)Jl>    4
^U-hiJ ^9>xllVI -Xj>JI 4
^Ab^oJJ o-'3VI -Jj>!-II 4
i^svjg>xiiyi yo^l&dl yS>.o 4
■~i I j nl -in II__fljgxmJI dfljjbJI _plgji || donl^ll dl~vn «g uljlanj ^punJI 9 plgjxll jLpI m
cLialjgJU uVg_^
English-Website | +9.
public@iuga2a.edu
```

Figure A. 6: The result of extracting the text in image was appearing in figure A.3.



○ Automatically select languages from the following list:

English; French; German; Italian; Spanish

● Specify languages manually

English;

⊟ Natural languages
- □ Abkhaz
- □ Adyghe
- □ Afrikaans
- □ Agul
- □ Albanian
- □ Altaic
- □ Armenian (Eastern) (Dictionary support)
- □ Armenian (Grabar) (Dictionary support)
- □ Armenian (Western) (Dictionary support)
- □ Avar
- □ Aymara
- □ Azerbaijani (Cyrillic)
- □ Azerbaijani (Latin)
- □ Bashkir (Dictionary support)
- □ Basque

Figure A. 7: supported language window

When tried Quick tasks which is the second process in ABBYY FineReader, we test some examples and include three samples with different situation, and found an excellent extraction for English text from image too. There are three cases. Case one, if image contained Arabic and English with white background, a conformation message appear because the resolution of image is less than 400 dpi and the program found characters cannot be recognized because the data set for Arabic characters does not include in the system to use it for character reorganization but detect an English characters or numbers, *see (figure A.10-A.13)*. So the result will appear numbers and characters of English but garbage for Arabic characters or numbers. The second case, if the image containing only Arabic text, the system cannot be processed and present a confirmation message it does not detect any character. So, the result is an image as you *see in (figure A.14-A.16).* The third case, if image contained Arabic text with format style, the program extracts it as picture, see (figure *A.13, A.19*).

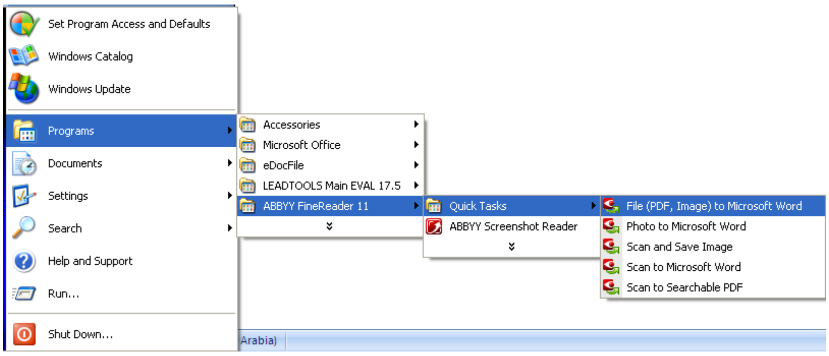
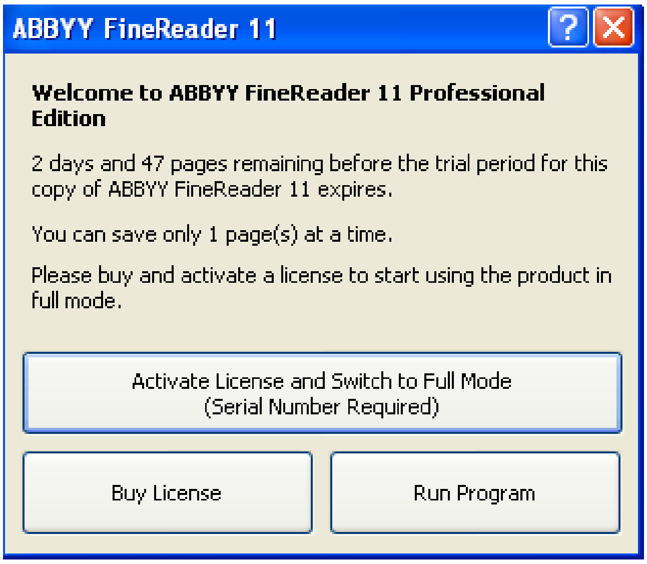


Figure A. 8: to run ABBYY Quick Tasks



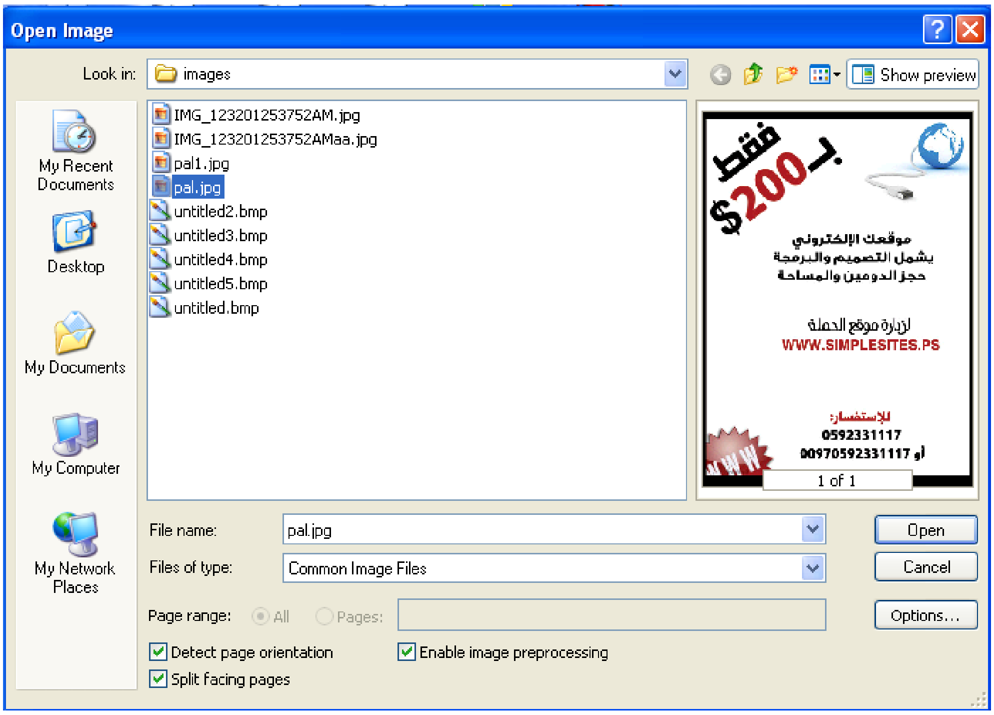
Welcome to ABBYY FineReader 11 Professional Edition

2 days and 47 pages remaining before the trial period for this copy of ABBYY FineReader 11 expires.

You can save only 1 page(s) at a time.

Please buy and activate a license to start using the product in full mode.

Figure A. 9: this window appear when you choose Quick Tasks in figure A.6



IMG_123201253752AM.jpg
IMG_123201253752AMaa.jpg
pal1.jpg
pal.jpg
untitled2.bmp
untitled3.bmp
untitled4.bmp
untitled5.bmp
untitled.bmp

File name: pal.jpg

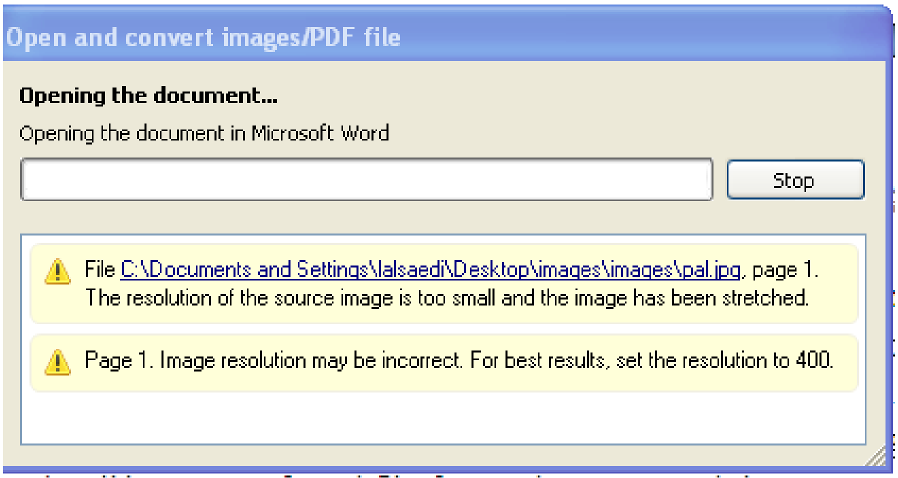
Files of type: Common Image Files

Page range: ● All ○ Pages:

☑ Detect page orientation ☑ Enable image preprocessing

☑ Split facing pages

Figure A. 10: the first example containing Arabic and English text with white background.



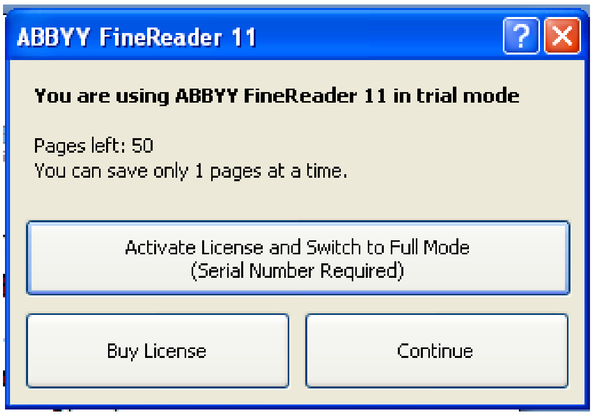
Opening the document...

Opening the document in Microsoft Word

File C:\Documents and Settings\lalsaedi\Desktop\images\images\pal.jpg, page 1. The resolution of the source image is too small and the image has been stretched.

Page 1. Image resolution may be incorrect. For best results, set the resolution to 400.

Figure A. 11: conformation message because resolution is less than 400.

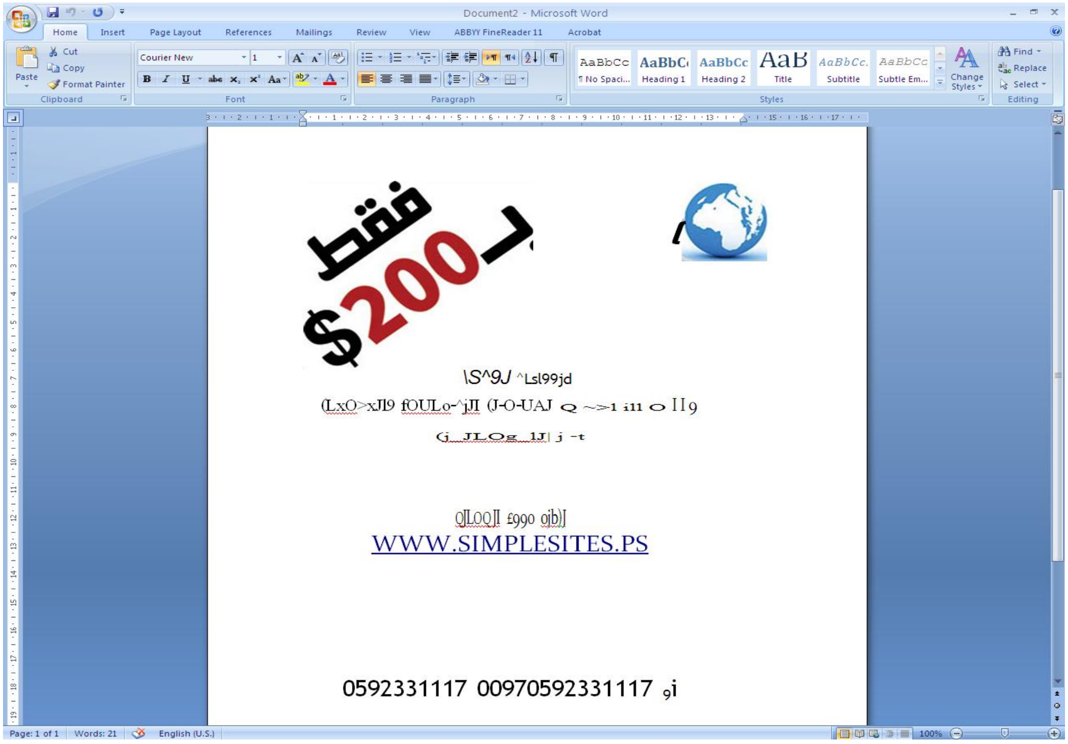


ABBYY FineReader 11

You are using ABBYY FineReader 11 in trial mode

Pages left: 50
You can save only 1 pages at a time.

Figure A. 12: this window appear to continue the convetion process



\S^9J ^Lsl99jd

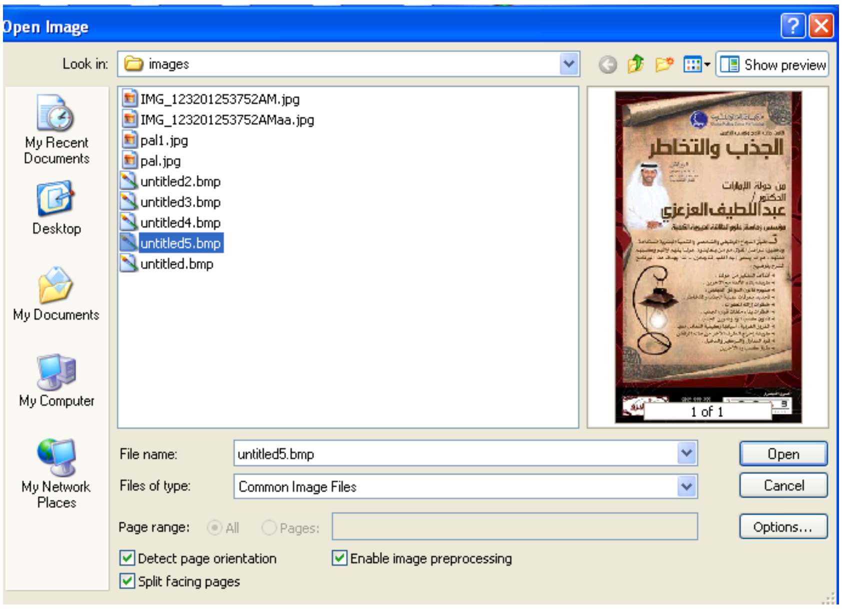
(LxO>xJl9 fOULo-^jJI (J-O-UAJ Q ~>1 i11 O II9

(j_JLOg_1J| j -t

QJLOQJI £990 ojb)J
WWW.SIMPLESITES.PS

0592331117 00970592331117 أو

Figure A. 13: the result for first example used in figure A.8



IMG_123201253752AM.jpg
IMG_123201253752AMaa.jpg
pal1.jpg
pal.jpg
untitled2.bmp
untitled3.bmp
untitled4.bmp
untitled5.bmp
untitled.bmp

File name: untitled5.bmp

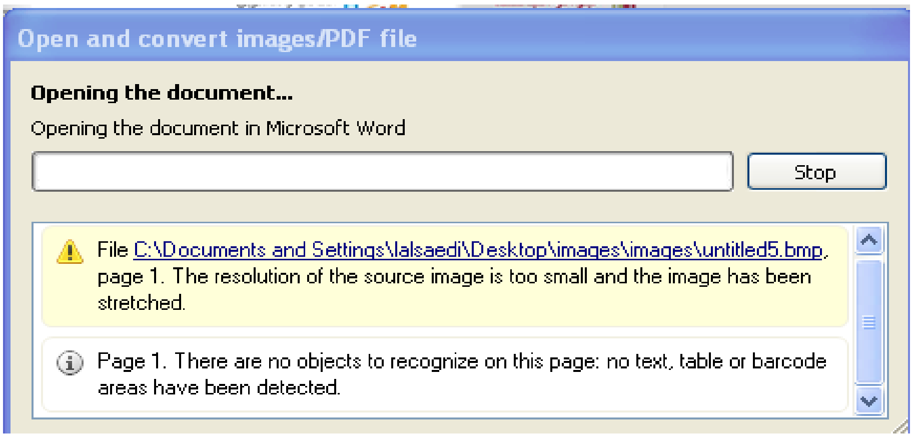
Files of type: Common Image Files

Page range: ● All ○ Pages:

☑ Detect page orientation ☑ Enable image preprocessing

☑ Split facing pages

Figure A. 14: the second example containing only Arabic text with background color.



Opening the document...

Opening the document in Microsoft Word

File C:\Documents and Settings\lalsaedi\Desktop\images\images\untitled5.bmp, page 1. The resolution of the source image is too small and the image has been stretched.

Page 1. There are no objects to recognize on this page: no text, table or barcode areas have been detected.

Figure A. 15: conformation message represent the program can not detect any object to recognize it.

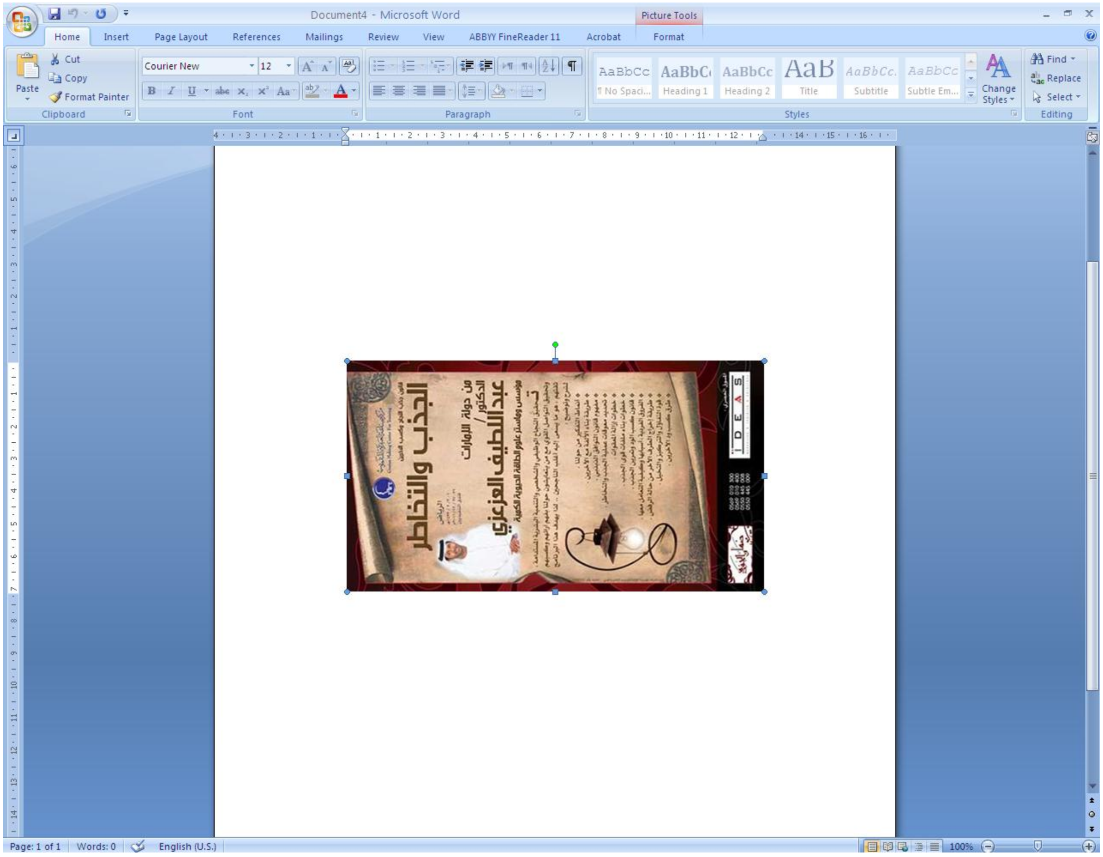
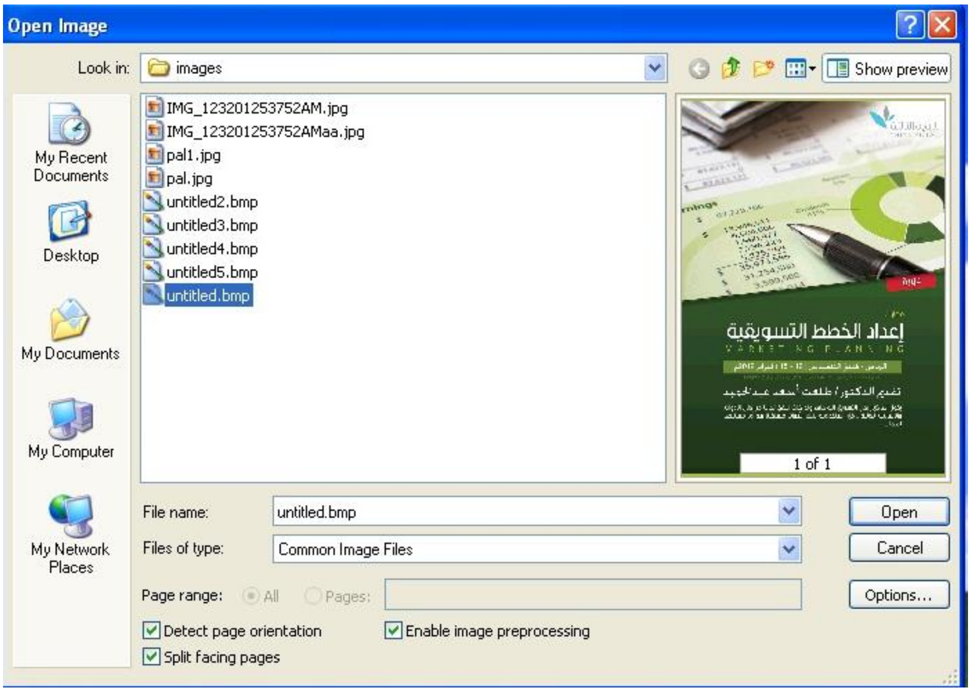


Figure A. 16: result for second example take in figure A.12.



Look in: images

IMG_123201253752AM.jpg
IMG_123201253752AMaa.jpg
pal1.jpg
pal.jpg
untitled2.bmp
untitled3.bmp
untitled4.bmp
untitled5.bmp
untitled.bmp

File name: untitled.bmp

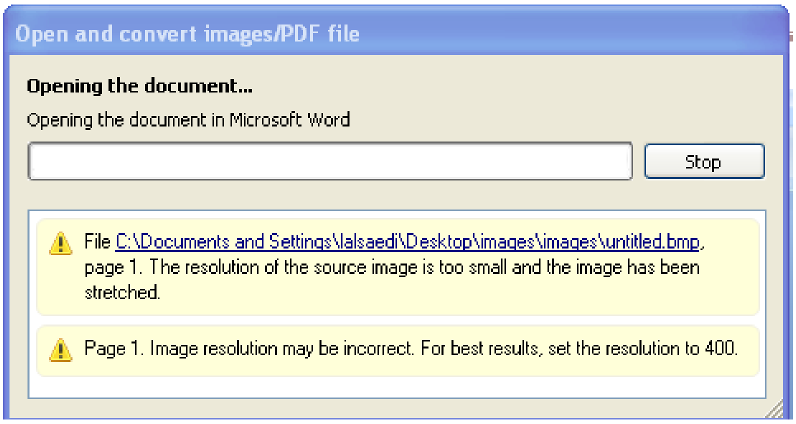
Files of type: Common Image Files

Page range: ○ All ○ Pages:

☑ Detect page orientation ☑ Enable image preprocessing

☑ Split facing pages

Figure A. 17: third example containing Arabic and English with background color.



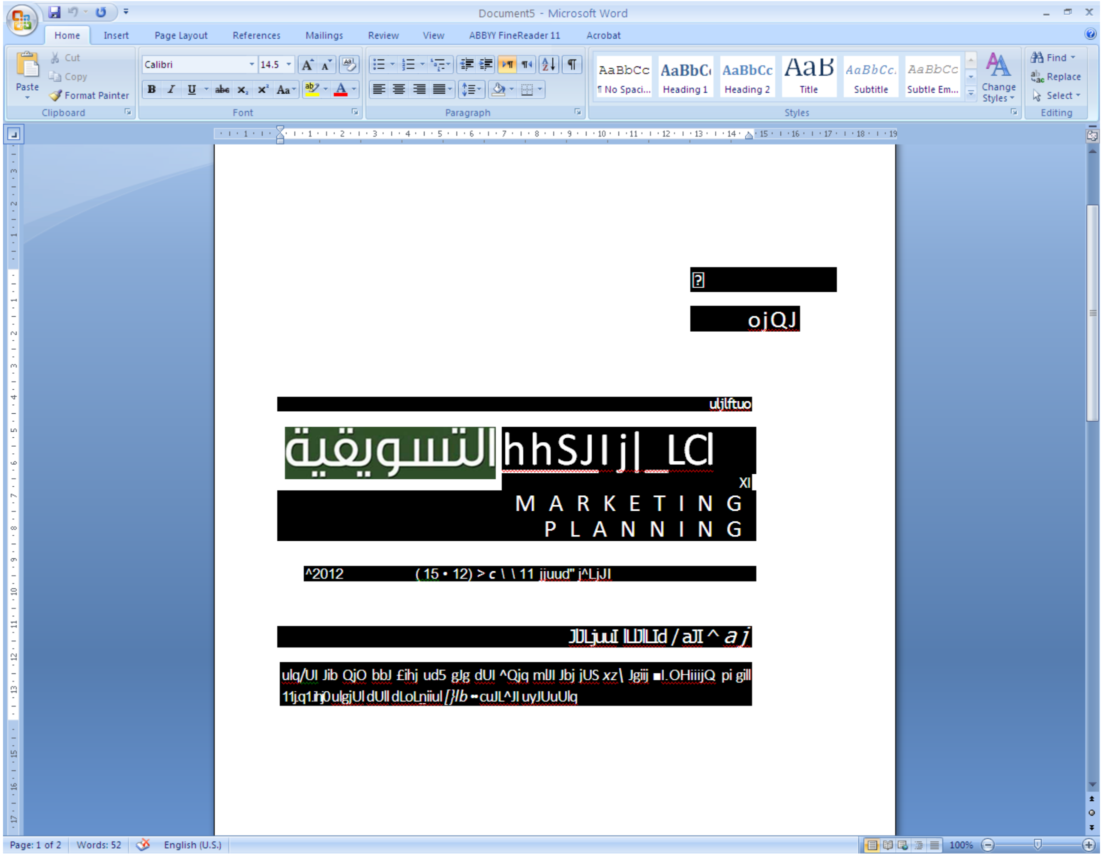
Opening the document...

Opening the document in Microsoft Word

File C:\Documents and Settings\lalsaedi\Desktop\images\images\untitled.bmp, page 1. The resolution of the source image is too small and the image has been stretched.

Page 1. Image resolution may be incorrect. For best results, set the resolution to 400.

Figure A. 18: conformation message because resolution is less than 400.



?

ojQJ

uljlftuo

التسويقية hhSJIj|_LCl

XI

MARKETING
PLANNING

^2012 (15 • 12) > c \ \ 11 jjuud" j^LjJI

JJLjuuI lLlJlLId / aJI ^ a j

ulq/UI Jib QjO bbJ £ihj ud5 gJg dUI ^Qjq mlJI Jbj jUS xz\ Jgiij ■I.OHiiijQ pi gill 11j.q1.ihj0ulgjUl dUll dLoLniiul [}lb ↔cuJL^Jl uyJUuUlq

Figure A. 19: result for third example take in figure A.15

Appendix B

Pseudo Code

In Appendix B we represent a pseudo code in MATLAB for each algorithm in chapter three.

Pseudo Code for Algorithm 3.1: threshold for edge map

```
Input: colored image I
Output: binary image

    1. X = grayscale image(I)
    2. S = wiener2(X)
    3. K = rangfilt(S)
    4. Level = graythresh(K)
    5. BW = convert grayscale image to black white with (K, Level)
```

Pseudo Code for Algorithm 3.2: Localization process step 1

(Collect information)

```
Input: Binary image BW
Output 1: localized each item in I via dashed rectangle shape
Output 2: ascending sort object_array by minimum row

    1. bw3 = bw-distance (BW)
    2. [L1  z] = bw-label (bw3)
       Note: L1 is the label number and z is  the total number of all labels
    3. Put any other elements in the image like background to zeros.
```

```
For j=1 to z
    [r, c]=find(L1==j)
    Object-array(j,1)=0
    Object_array(j,2)=j
    Object_array(j,3)= Maximum_row
    Object_array(j,4)= Minimum_row
    Object_array(j,5)= Maximum_column
    Object_array(j,6)= Minimum_column
    Width = object_array(jjj,5)-object_array(jjj,6)
    Height = object_array(jjj,3)-object_array(jjj,4)
    If (width>0) and (height>0)
        Draw rectangle shape with these parameter (Mnc, Mnr, width, height, line width, line style)
    End
End
Sort object_array in ascending order depending on min_row.
```

Pseudo Code for Algorithm 3.3: compute difference and mean

Input 1: ascending sort object_array by minimum row use Pseudo Code for algorithm 3.2

Input 2: z number of items

Input 3: kind variable, it maybe max or min row or col.

Output 1: mean for input 1

Output 2: difference_array1 for input1

```
Difference__array (z)= 0
sum = 0
For ost from 2 to z ⟶          z is the number of items
    Difference_array (ost) = object_array (ost, kind) - object_array(ost - 1, kind)
    sum = sum + Difference_array (ost)
```

```
End
Avg = └ mean(Difference_array) ┘
```

Pseudo Code for Algorithm 3.4: Localization process step 2

(give label for each new row)

Input 1: ascending sort object_array by minimum row use algorithm 3.2

Input 2: difference array of min row (see Pseudo Code for algorithm 3.3)

Input 3: average of differences of min row (see Pseudo Code for algorithm 3.3)

Output 1: ascending sort object_array by minimum column

```
Object_array (1,1)= 1
Label = 1
Obj (z) = 0
Xl = 0
For ph from 2 to z        →        z is the number of items
    If(Difference_minrow(ph) > Avg_minrow )
        Label= label+1
        Object_array (ph,1)=label
        Xl = Xl +1          }
            Obj(Xl) = ph    }  →  To save the place of each new label and use it as a range in sort by min column
        End
    End
Sort (object-array, Obj, Xl, min_col)   →  sort elements in range of each label in ascending order depending on min_column.
```

Pseudo Code for Algorithm 3.5: Localization process step 3

(give labels for the rest of items in object_array)

Input 1: ascending sort object_array by minimum row and column use Pseudo Code for algorithm 3.4.

Input 2: Avgmaxrow is average of max row (see Pseudo Code for algorithm 3.3)

Input 3: Avgmincol is average of min column (see Pseudo Code for algorithm 3.3)

Output 1: sort object_array by labels in ascending order

```
For fg from 2 to z  →  z is the number of items
Fc = fg - 1
    While (object_array(fg,1)==0 )
        If (| object_array(fg, maxrow)- object_array(fc, maxrow) <= Avgmaxrow |)
            If (| object_array(fg, mincol)- object_array(fc, mincol) <= Avgmincol |)
                Object_array(fg,1)=Object_array(fc,1)
            Else
                Label= label+1
                Object_array (fg,1)=label
            End
        Else
            Label= label+1
            Object_array (fg,1)=label
            End
        End
Sort (object-array, label)  →  sort all elements in object_array by label in ascending order.
```

Pseudo Code for Algorithm 3.6: Localization process step 4

(find minimum and maximum row and column after new label classification)

Input 1: ascending sort object_array by label use Pseudo Code for algorithm 3.5.

Output 1: start_min_row vector is a vector that contains all minimum rows for after classification labels

Output 2: start_min_col vector is a vector that contains all minimum columns after new classification labels

Output 3: end_max_row vector is a vector that contains all maximum rows after new classification labels

Output 4: end_max_col vector is a vector that contains all maximum columns after new classification labels

```
Searchforlabel = 1
Px = 1
While (Searchforlabel <= z) and (Px <= z)
    start_min_row (Searchforlabel) = object_array (Px, minrow)
    start_min_col (Searchforlabel)= object_array (Px, mincol)
    end_max_row (Searchforlabel)= object_array (Px, maxrow)
    end_max_col (Searchforlabel)= object_array (Px, maxcol)
    For similar from Px+1 to z
        If (object_array(similar,1)== Searchforlabel)
                Px = Px+1
            If (object_array (similar, minrow) <= start_min_row (Searchforlabel))
                start_min_row (Searchforlabel) = object_array (similar, minrow)
            End
            If (object_array (similar, mincol) <= start_min_col (Searchforlabel))
            start_min_col (Searchforlabel) = object_array (similar, mincol)
          End
          If (object_array (similar, minrow) <= end_min_row (Searchforlabel))
            end_max_row (Searchforlabel) = object_array (similar, maxrow)
          End
          If (object_array (similar, minrow) == start_min_row (Searchforlabel))
            end_max_col (Searchforlabel) = object_array (similar, maxcol)
          End
      End
    End
          Searchforlabel = Searchforlabel+1
          Px = Px+1
End
```

Pseudo Code for Algorithm 3.7: Localization process step 5

(select text region)

Input 1: start_min_row vector is a vector that contains all minimum rows for after classification labels use Pseudo Code for algorithm 3.6.

Input 2: start_min_col vector is a vector that contains all minimum columns after new classification labels use Pseudo Code for algorithm 3.6.

Input 3: end_max_row vector is a vector that contains all maximum rows after new classification labels use Pseudo Code for algorithm 3.6.

Input 4: end_max_col vector is a vector that contains all maximum columns after new classification labels use Pseudo Code for algorithm 3.6.

Output 1: vector of text region crop

Output 2: obj,region number

Output 3, 4, 5, 6:start_min_row, start_min_col, end_max_row, end_max_col vectors

```
For obj from 1 to label
widthRegion = end_max_col (obj) -  start_min_col (obj)
heightRegion = end_max_row (obj) -  start_min_row (obj)
 if (widthRegion>0) and (heightRegion>0)
        draw rectangle a round (start_min_col (obj), start_min_row (obj), widthRegion, heightRegion )
        [ImageCrop] = crop region on (black and white image) with the same parameters of rectangle shape
 End
End
```

Pseudo Code for Algorithm 3.8: collect information

Input: BW3 region of Binary image crop

Output: full-info-matrix is a Tow dimension array

```
[zx  zy]= size(BW3)
Full-Info-matrix (zx,3)= 0
img-row = 1
    For i = 1 to zx
        B-count=0
        W-count=0
         For j = 1 to zy
             Full-Info-matrix (j, 1)= img-row
             If (BW3 (i, j) == 0)
                  B-count =B-count+1
            Else
                  W-count =W-count+1
            End
        Full-Info-matrix (j, 2)= B-count
        full-Info-matrix (j, 3)= W-count
        Img-row=Img-row+1
    End
```

Pseudo Code for Algorithm 3.9: Calculation Arabic Base Line

Input: two dimension array full-info-img use Pseudo Code for algorithm 3.8.

Output: The Base Line Number

```
maxm = full-info-img(1,3)
Base = 1
For mp = 2 to imx1
    If (maxm < full-info-img (mp, 3))
        maxm = full-info-img (mp, 3)
        Base = mp
    end
End
```

Pseudo Code for Algorithm 3.10: find lengths of adjacent sequence black pixels in Base line

Input: Base line for BW3 region of Binary image crop use Pseudo Code for algorithm 3.9.

Output: descending Bline vector, this vector contains length of adjacent sequence black pixels in base line

```
[zx  zy]= size(BW3)
x=1
For m from 1 to zy
    Bline (m)= 0
```

```
End
    sumb = 1
    For L from 1 to zy-1
            If (BW3 (Base, L) == 0) and (BW3 (Base, L+1) == 0)
                sumb =sumb +1
            End
            If (BW3 (Base, L) == 0) and (BW3 (Base, L+1) == 1)
                Bline(x) =sumb
                x = x + 1
                Sumb = 1
            End
      End
    Descending sort for Bline
```

Pseudo Code for Algorithm 3.11: find accumulator for lengths of adjacent sequence black pixels in Base line

Input 1: descending Bline vector, this vector contains length of adjacent sequence black pixels in base line use Pseudo Code for algorithm 3.10.

Input 2: x value, where x is length of Bline vector

Output: accum(x, 2) is a two dimension accumulator, first column is the length value and the second column is how many times this value redundant in Bline vector

```
accum(x,2)= 0
bc = 1
b = 1
sumc = 1
Bfinish = 0
For m from 1 to x
    accum (m,1)= 0
    accum (m,2)= 0
End
```

```
While(b < = x)
     If (Bline (b) == 0)
                 b = b +1
     Else
           If (not Bfinish)
                 bb = Bline(b)
                 accum(bc,1) =bb
                 Bfinish = 1
                 sumc = 0
           End
           If (Bline(b)  != bb)
              accum(bc,2) = sumc
              bc = bc + 1
              bb= Bline(b)
              accum(bc,1) = bb
              sumc = 1
           End
   b = b +1
   End
End
If (b > x)
  accum(bc,2) = sumc
End
```

Pseudo Code for Algorithm 3.12: find peak and its position

Input 1: accum(x, 2) is a two dimension accumulator, first column is the length value and the second column is how many times this value redundant in Bline vector use Pseudo Code for algorithm 3.11.

Input 2: bc value, where bc is length of accum two dimension array

Output 1: maximum peak value

Output 2: C, the position of maximum peak

```
peak = 1
```

```
C = 1
For ui from 1 to bc -1
    peak = accum (ui,2)
    C= ui
    For uj from ui+1 to bc
        If (peak < accum(uj,2))
            peak = accum (uj,2)
            C = uj
        End
    End
End
```

Pseudo Code for Algorithm 3.13: find half line and Base line ratio

Input 1: imx, is the height region crop

Input 2: Base value for the region crop use Pseudo Code for algorithm 3.9.

Output 1: halfline, the value of half line

Output 2: ratio, range that maybe found place of base line in the region crop

```
halfline = └ imx / 2┘
defhalf = imx – halfline
defBase = imx - Base
    If (defhalf < defBase)
        ratio = └ defhalf / defBase ┘
    Else
        ratio = └ defBase / defhalf ┘
    End
ratio = (ratio * Base)
```

Distance between region height and half line

Distance between region height and Base line

Pseudo Code for Algorithm 3.14: Set of Rules

Input 1: objarea, is the ratio of foreground area to the region area

Input 2: Base, the value of base line

Input 3: halfline, the value of half line

Input 4: ratio, the range could be found the base line below the half line

Input 5: accum two dimension array, includes in first column the times of strait line of black pixels in the base line are redundant

Input 6: val, the value in the second column of accum array

Input 7: C, the position in accum that contain maximum peak value

Input 8: start_min_row, start_min_col, end_max_row, end_max_col vectors, these vectors contains the origin coordinates from the origin image

Input 9: BW3, thinning image crop

Input 10: I2, blank BW image with white background and has the same size of original RGB image

Output: I2 output result image, this include Arabic text with some unwanted data

```
Apply Pseudo Code for algorithm 3.7 (select text region) to return obj, start_min_row, start_min_col, end_max_row, and end_max_col.
Convert white background to black
BW3 = thinning to image_crop
Farea = Find foreground area
Imagearea = Find image_crop area (width * height)
Objarea = Farea / Imagearea
Apply Pseudo Code for algorithm 3.13 to find halfline and ratio
Apply Pseudo Code for algorithm 3.11 to find val and C
Size (I2)=size (I) ⟶          I is the origin image
I2 = 1 ⟶          i.e.: I2 has white background
if (Objarea >= 0.01)and(Objarea <= 0.16)
    if (Base >= halfine) and (Base <= hafline + ratio)
        if((accum (C, 1) >= 1) and (val >= 3))
            ixc = 1
             For hr from start_min_row(obj) to end_max_row(obj)
               iyc = 1
```

```
17.                 For hc from start_min_col(obj) to end_max_col(obj)
18.                     If (hr != 0) and (hc != 0)
19.                           I2(hr, hc) = BW3(ixc, iyc)
20.                     End
21.                     iyc= iyc +1
22.                 End
23.                ixc = ixc + 1
24.            End
25.       End
26.    End
27.End
```

The variable ixc and iyc are using to make a copy from the region image crop to its position in I2 output image.

Pseudo Code for Algorithm 3.15: OCR Post Processing

Input: I2 output result image, this include Arabic text with some unwanted data use Pseudo Code for algorithm 3.13

Output: I3 output image to OCR system

```
1.  Apply Pseudo Code for algorithm 3.2and return clip region maxr,
    maxc, minr, and minc. but with change the distance with 4 and
    without use objectarray and ascending sort
2.  [dx dy] = size(clip)

3.  BWarea = foreground object area
4.  cliparea = dx * dy ⟶        (width * height)
5.  blackratio = 2/3 * cliparea
6.  countwhiteratioarea=countwhite / cliparea
7.  countblackratioarea=countblack / cliparea
8.  blackoverwhite=countblack / countwhite
9.  if (countblackratioarea <= 0.28)
10.    if (BWarea > 30)
11.        if(blackratio / BWarea >= 2.5)
12.            If (blackoverwhite < 0.251)
13.           ixc = 1
14.            For hr from minr to maxr)
```

```
            iyc = 1
              For hc from minc to maxc
                  If (hr != 0) and (hc != 0)
                      I3(hr, hc) = BW3(ixc, iyc)
                  End
                  iyc= iyc +1
              End
            ixc = ixc + 1
          End
        End
      End
    End
End
```

The variable ixc and iyc are using to make a copy from the region image crop to its position in I2 output image.

Pseudo Code for Algorithm 3.16: Show Post Processing

Input: I, is a colored image

Output: Img, is output image for show

```
I2 = convert to binary of I
I3 =  preprocessing ( I )
I4 = postprocessing ( I3 )
Swap between black and white in I4
Now apply thicken skeleton on I4
Swap again between black and white in I4
rgb = Convert I4 image to RGB image
I2 = Convert I2 image to RGB image
Change foreground color in I4 to red color
Img = multiply I2 with rgb
```

I3 and I4 is binary image

Appendix C

Image Groups

Table C. 1: group A includes different style texts were written on white background

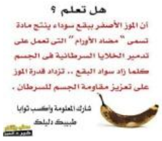
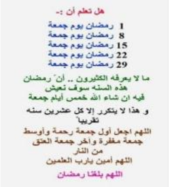
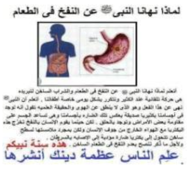
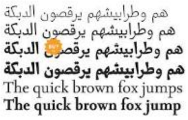
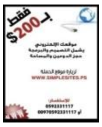
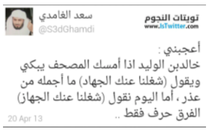
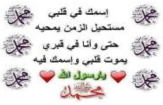
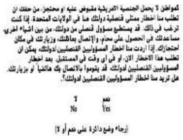
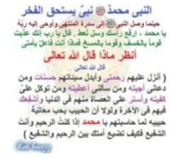
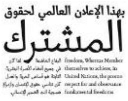
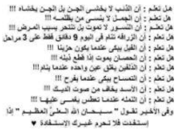
Group A				
1.  هل تعلم ؟ أن الموز الأصفر ببقع سوداء ينتج مادة تسمى " مضاد الأورام " التى تعمل على تدمير الخلايا السرطانية فى الجسم كلما زاد سواد البقع .. تزداد قدرة الموز على تعزيز مقاومة الجسم للسرطان . شارك المعلومة واكسب ثوابا طبيبك دليلك correlation: in=1 out=0	2.  هل تعلم أن :- 1 رمضان يوم جمعة 8 رمضان يوم جمعة 15 رمضان يوم جمعة 22 رمضان يوم جمعة 29 رمضان يوم جمعة ما لا يعرفه الكثيرون .. أن" رمضان هذه السنه سوف نعيش فيه ان شاء الله خمس أيام جمعة و هذا لا يتكرر إلا كل عشرين سنه تقريباً اللهم اجعل أول جمعة رحمة وأوسط جمعة مغفرة واخر جمعة العتق من النار اللهم أمين يارب العلمين اللهم بلغنا رمضان correlation: in=0.89 out=0.1	3.  لماذا نهانا النبى ﷺ عن النفخ فى الطعام هذه سنة نبيكم علِم الناس عظمة دينك أنشرها correlation: in=0.91 out=0.08	4. يبقي حد يقول خروج بعد كده كل خروجة ينكد علينا!!! دي كانت خروجة سوووووووووووووا correlation: in=0.84 out=0.15	5.  هم وطرابيشهم يرقصون الدبكة هم وطرابيشهم يرقصون الدبكة هم وطرابيشهم يرقصون الدبكة هم وطرابيشهم يرقصون الدبكة The quick brown fox jumps The quick brown fox jump correlation: in=0.98 out=0.018
6.  فقط بـ200$ موقعك الإلكتروني يشمل التصميم والبرمجة حجز الدومين والمساحة لزيارة موقع الحملة WWW.SIMPLESITES.PS للإستفسار: 0592331117 أو 00970592331117 correlation: in=0.99 out=0.0004	7. ياربّ ، إستودعتك دعواتي فبشرني بها correlation: in=1 out=0	8. correlation: in=0.96 out=0.032	9. بجد!! correlation: in=0.94 out=0.03	10.  سعد الغامدي @S3dGhamdi تويتات النجوم www.IsTwitter.com أعجبني : خالدبن الوليد اذا أمسك المصحف يبكي ويقول (شغلنا عنك الجهاد) ما أجمله من عذر ، أما اليوم نقول (شغلنا عنك الجهاز) الفرق حرف فقط .. 20 Apr 13 correlation: in=0.75 out=0.24
11.  إسمك في قلبي مستحيل الزمن يمحيه حتى وأنا في قبري يموت قلبي وإسمك فيه يارسول الله محمد correlation: in=0.99 out=0.001	12.  لا نعم No Yes (رجاء وضع دائرة على نعم أو لا) correlation: in=0.87 out=0.12	13.  النبى محمدٌ ﷺ نبىٌ يستحق الفخر أنظر ماذا قال الله تعالى correlation: in=0.92 out=0.071	14. correlation: in=0.82 out=0.17	15. correlation: in=0.77 out=0.22
16.  بهذا الإعلان العالمي لحقوق المشترك correlation: in=0.99 out=0.0012	17.  هل تعلم : أن الذئب لا يخشى الجن بل الجن يخشاه !!! هل تعلم : أن الجمل لا ينسى من يظلمه !!! هل تعلم : أن النسور لا تموت بل تنتحر بسبب المرض !!! هل تعلم : أن الزرافه تنام فى اليوم 9 دقائق فقط على 3 مراحل هل تعلم : أن الفيل يبكي عندما يكون حزينا !!! هل تعلم : أن الحصان يموت إذا قطع ذيله !!! هل تعلم : أن الدلفين يغلق عين واحده عندما ينام !!! هل تعلم : أن التمساح يبكي عندما يفرح !!! هل تعلم : أن الأسد يخاف من صوت الديك !!! هل تعلم : أن النمله عندما تعطس يغمى عليها !!! وفى الأخير نقول " سبحان الله العلىّ العظيم " إذا إستفدت فلا تحرم غيرك الإستفادة ♥ correlation: in=0.97 out=0.024	18. اكيد انت في فلسطين correlation: in=1 out=0	19. ربِّ اجْعَلْ أُمِّي سَيِّدَةً مِن سَيِّدَاتِ أهْلِ الْجَنَّةِ correlation: in=0.73 out=0.26	

Table C. 2: group B includes different style texts were written on background filled with one color rather than white.

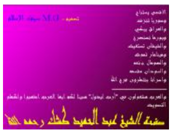
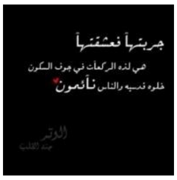
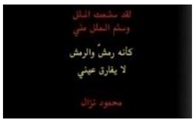
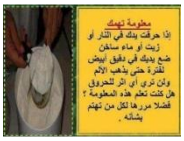
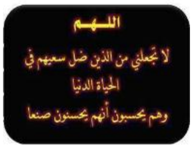
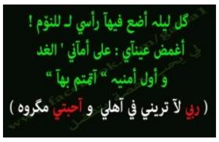
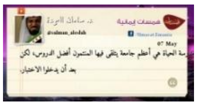
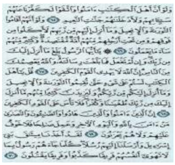
Group B				
1.  correlation: in=1 out=0	2.  جربتها نعشقتها هي لذه الركعات في جوف السكون خلوه تدسيه والناس نائمون correlation: in=1 out=0	3. هي يا سجاني هي يا عَتمة الزنزانةِ correlation: in=0.96 out=0.034	4. correlation: in=1 out=0	5.  correlation: in=1 out=0
6. وَإِذَا سَأَلَكَ عِبَادِي عَنِّي فَإِنِّي قَرِيبٌ أُجِيبُ دَعْوَةَ الدَّاعِ إِذَا دَعَانِ correlation: in=0.99 out=0.035	7.  معلومة تهمك إذا حرقت يدك في النار أو زيت أو ماء ساخن ضع يديك في دقيق أبيض لفترة حتى يذهب الألم ولن ترى أي اثر للحروق هل كنت تعلم هذه المعلومة ؟ فضلا مررها لكل من تهتم بشأنه . correlation: in=0.96 out=0.037	8.  اللـــهـــم لا تجعلني من الذين ضل سعيهم في الحياة الدنيا وهم يحسبون أنهم يحسنون صنعا correlation: in=0.81 out=0.18	9. اللهم أرزق جميع بنات المسلمين بالزوج الصالح الذي تحبه وترضاه correlation: in=0.99 out=0.001	10.  گل ليله أضع فيهآ رأسي لـ للنوْم ! أغمض عينآي : على أمآني ' الغد و أول أمنيه " آتمتم بهآ " (ربي لآ تريني في آهلي و آحبتي مكروه) correlation: in=0.98 out=0.016
11. correlation: in=1 out=0	12. correlation: in=0.87 out=0.12	13.  correlation: in=1 out=0	14.  correlation: in=0.92 out=0.073	15. correlation: in=1 out=0

Table C. 3: group C includes different style texts were written on background filled with variant/gradient color

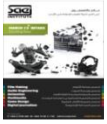
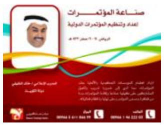
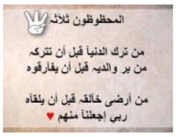
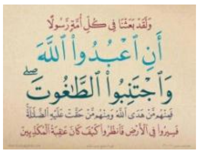
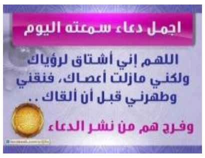
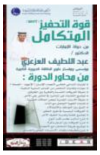
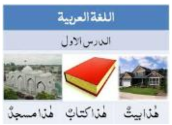
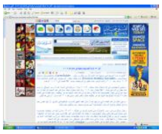
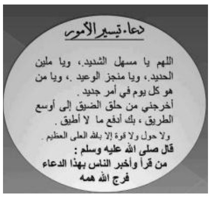
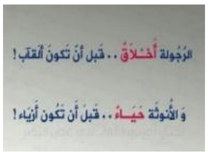
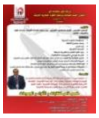
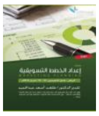
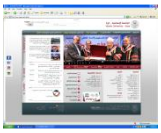
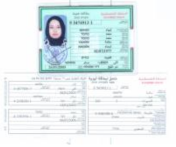
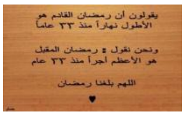
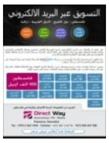
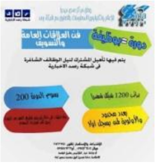
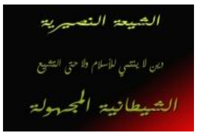
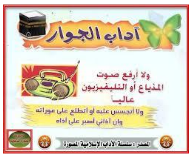
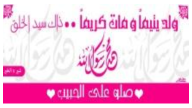
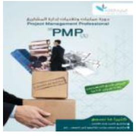
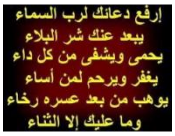
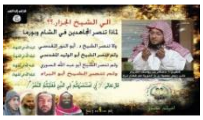
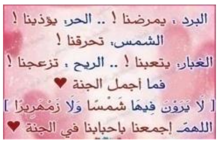
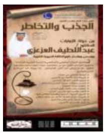
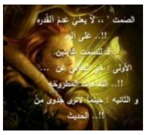
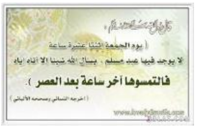
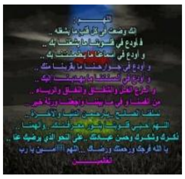
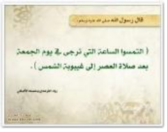
Group C				
1.  correlation: in=0.99 out=0.0005	2.  correlation: in=0.99 out=0.0005	3.  المحظوظون ثلاثة من ترك الدنيا قبل أن تتركه من بر والديه قبل أن يفارقوه من أرضى خالقه قبل أن يلقاه ربي إجعلنا منهم ♥ correlation: in=0.96 out=0.31	4.  وَلَقَدْ بَعَثْنَا فِي كُلِّ أُمَّةٍ رَّسُولًا أَنِ اعْبُدُوا اللَّهَ وَاجْتَنِبُوا الطَّاغُوتَ فَمِنْهُم مَّنْ هَدَى اللَّهُ وَمِنْهُم مَّنْ حَقَّتْ عَلَيْهِ الضَّلَالَةُ فَسِيرُوا فِي الْأَرْضِ فَانظُرُوا كَيْفَ كَانَ عَاقِبَةُ الْمُكَذِّبِينَ correlation: in=0.86 out=0.13	5.  اجمل دعاء سمعته اليوم اللهم إني أشتاق لرؤياك ولكني مازلت أعصاك، فنقني وطهرني قبل أن ألقاك .. وفرج هم من نشر الدعاء correlation: in=0.83 out=0.16
6.  correlation: in=0.72 out=0.27	7.  اللغة العربية الدرس الاول هٰذا بيتٌ هٰذا كتابٌ هٰذا مسجدٌ correlation: in=1 out=0	8.  correlation: in=0.92 out=0.072	9.  دعاء تيسير الأمور اللهم يا مسهل الشديد، ويا ملين الحديد، ويا منجز الوعيد ، ويا من هو كل يوم في أمر جديد . أخرجني من حلق الضيق إلى أوسع الطريق ، بك أدفع ما لا أطيق . ولا حول ولا قوة إلا بالله العلى العظيم . قال صلى الله عليه وسلم : من قرأ وأخبر الناس بهذا الدعاء فرج الله همه correlation: in=0.92 out=0.073	10.  الرُّجُولة أَخـلاَق .. قَبل أنّ تَكونَ ألقَاب ! وَ الأُنوثَة حَيَـاء .. قَبلَ أن تَكُون أزيَاء ! correlation: in=0.95 out=0.04
11.  correlation: in=0.96 out=0.038	12.  correlation: in=0.82 out=0.17	13.  correlation: in=0.84 out=0.15	14.  correlation: in=0.84 out=0.15	15.  يقولون أن رمضان القادم هو الأطول نهاراً منذ ٣٣ عاماً ونحن نقول : رمضان المقبل هو الأعظم أجراً منذ ٣٣ عام اللهم بلغنا رمضان ♥ correlation: in=0.93 out=0.066
16.  correlation: in=0.88 out=0.11	17.  correlation: in=0.99 out=0.004	18.  الشيعة النصيرية دين لا ينتمي للأسلام ولا حتى التشيع الشيطانية المجهولة correlation: in=0.77 out=0.22	19.  آداب الجوار ولا أرفع صوت المذياع أو التليفيزيون عاليـاً ولا أتجسس عليه أو أتطلع على عوراته وان آذاني أصبر على أذاه المصدر : سلسلة الآداب الإسلامية المصورة correlation: in=1 out=0	20.  ولد يتيماً و مات كريماً •• ذاك سيد الخلق محمد رسول الله ♥صلو على الحبيب♥ correlation: in=0.81 out=0.18
21.  correlation: in=0.97 out=0.023	22.  إرفع دعائك لرب السماء يبعد عنك شر البلاء يحمى ويشفى من كل داء يغفر ويرحم لمن أساء يوهب من بعد عسره رخاء وما عليك إلا الثناء correlation: in=0.82 out=0.17			

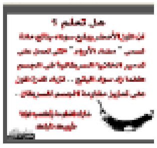
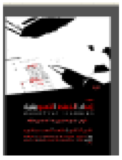
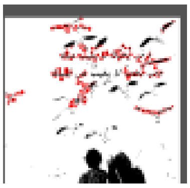
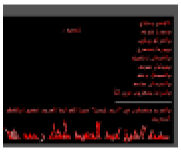
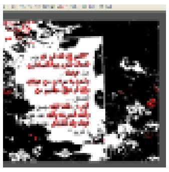
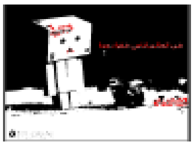
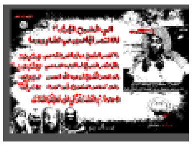
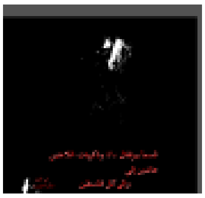
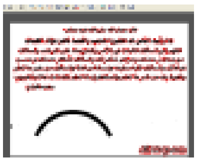
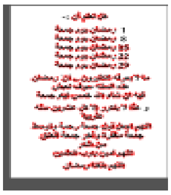
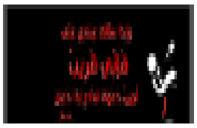
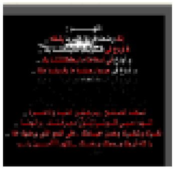
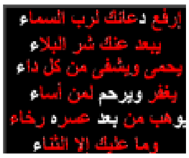
Table C. 4: group D includes different style texts were written on picture background

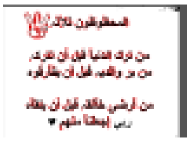
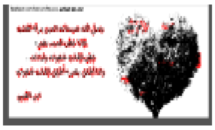
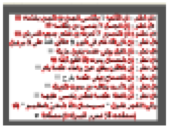
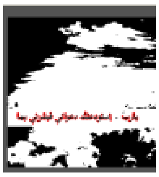
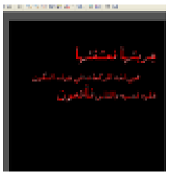
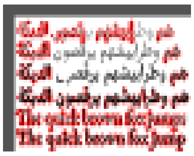
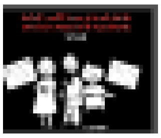
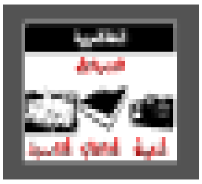
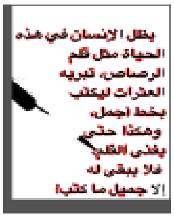
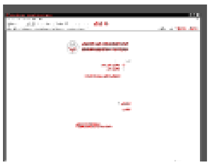
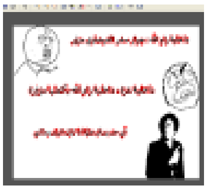
Group D				
1.  correlation: in=0.99 out=0.0009	2. لقد رأيت رجلا يتقلب في الجنة في شجرة قطعها من ظهر الطريق كانت تؤذي الناس correlation: in=0.92 out=0.075	3. correlation: in=0.79 out=0.2	4. حتى كوفيتك يا سيدي لعشقها لفلسطين رسمت نفسها على كتفيك correlation: in=0.88 out=0.11	5.  البرد ، يمرضنا ! .. الحر، يؤذينا ! الشمس، تحرقنا ! الغبار، يتعبنا ! .. الريح ، تزعجنا ! فما أجمل الجنة [لا يرون فيها شمساً ولا زمهريرا] اللهمّ إجمعنا بأحبابنا في الجنة correlation: in=0.98 out=0.017
6.  correlation: in=0.86 out=0.13	7. رَبِّ اجْعَلْنِي مُقِيمَ الصَّلَاةِ وَمِن ذُرِّيَّتِي رَبَّنَا وَتَقَبَّلْ دُعَاءِ correlation: in=1 out=0	8.  correlation: in=0.92 out=0.071	9. correlation: in=0.99 out=0.004	10. اللهم صل وسلم وبارك على سيدنا وحبيبنا محمد correlation: in=1 out=0
11. correlation: in=0.91 out=0.89	12. correlation: in=1 out=0	13. correlation: in=1 out=0	14. correlation: in=1 out=0	15. correlation: in=1 out=0
16. correlation: in=0.84 out=0.15	17. correlation: in=0.99 out=0.0077	18. correlation: in=0.87 out=0.12	19. correlation: in=0.98 out=0.014	20.  correlation: in=0.99 out=0.001
21. correlation: in=0.89 out=0.1	22. correlation: in=0.79 out=0.2	23. يقول ابن القيم رحمه الله ذنوب الخلوات اصل اسباب الانتكاسات الثبات correlation: in=0.68 out=0.31	24.  correlation: in=0.88 out=0.11	25.  correlation: in=0.89 out=0.1

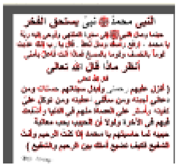
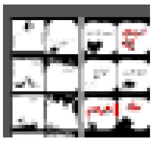
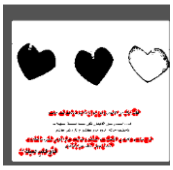
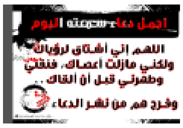
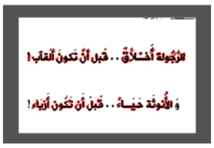
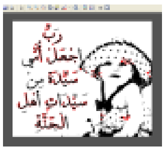
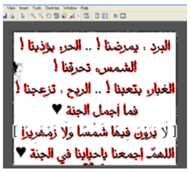
26. correlation: in=0.9 out=0.098	27. يظل الإنسان في هذه الحياة مثل قلم الرصاص، تبريه العثرات ليكتب بخط أجمل، وهكذا حتى يفنى القلم، فلا يبقى له إلا جميل ما كتب! correlation: in=0.95 out=0.046	28. إني أحبك حباً ليس يبلغه فهمي ولا ينتهي وصفي الى صفته اقصى نهاية علمي فيه معرفتي بالعجز مني عن ادراك معرفته correlation: in=0.57 out=0.42	29. أربعة أشخاص أحترمهم في حياتي - شخص وضع بي الثقة http://www.facebook.com/DuAlym - شخص أخلص لي بالوعد - شخص يقدر معنى الصداقة - شخص لم يقطع صلته بي رغم تقصيري معه correlation: in=0.79 out=0.2	30. correlation: in=1 out=0
31. correlation: in=0.97 out=0.027	32. correlation: in=1 out=0	33. أنثرو ورود السعادة أينما ذهبتم فالدنيا فانية لا تملؤوها بالأحزان وإياكم أن تيأسو من رحمة الله correlation: in=0.93 out=0.06	34. correlation: in=0.99 out=0.005	

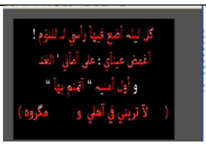
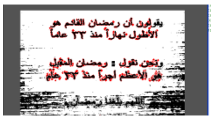
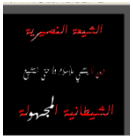
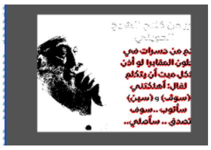
In the next table, we present the original image but in black white mode by marking the discovered text with red color.

Table C. 5: set of results images for show.

1. 	2. 	3. 	4.
5. 	6. 	7. 	8.
9. 	10. 	11. 	12. 
13. 	14. 	15. 	16. 
17. 	18. 	19. 	20. 
21. 	22. 	23.  إرفع دعائك لرب السماء يبعد عنك شر البلاء يحمي ويشفي من كل داء يغفر ويرحم لمن أساء يهب من يعط عمره رخاء وما عليك إلا الثناء	24.

25.  المحظوظون ثلاثة من ترك الدنيا قبل أن تتركه، من أرضى خالقه قبل أن يلقاه	26. 	27. علم الناس عظمة دينك انشرها	28.
29. 	30. 	31. 	32.
33. 	34. 	35.  The quick brown fox jump The quick brown fox jump	36.
37. 	38. 	39. 	40.
41. 	42. المشترك	43.  يظل الإنسان في هذه الحياة مثل قلم الرصاص، تبريه العثرات ليكتب بخط اجمل، وهكذا حتى يفنى القلم، فلا يبقى له إلا جميل ما كتبه!	44. 
45. 	46. 	47. 	48. 
49. أكيد أنت في فلسطين	50. 	51. 	52.

53. 	54. 	55. 	56.
57. 	58. 	59. 	60.
61. 	62. 	63. 	64.
65. 	66. 	67. 	68.
69. 	70. 	71.  أجمل دعاء سمعته اليوم اللهم إني أشتاق لرؤياك ولكني مازلت أعصاك، فنقني وطهرني قبل أن ألقاك وفرج هم من نشر الدعاء	72.  الرُّجولة أَخلاق .. قبل أن تكون ألقاب! والأُنوثة حَيـاء .. قبل أن تكون أزياء!
73. 	74. 	75. 	76.
77. 	78. 	79. 	80.
81.	82.	83.	84.

 اللـهـم لا تجعلني من الذين ضل سعيهم في الحياة الدنيا وهم يحسبون أنهم يحسنون صنعا	 كل ليله أضع فيها رأسي لـ للنوم ! أغمض عيناي : على آماني ' الغد و أول أمنيه " أتمتم بهآ " (لآ تريني في آهلي و مگروه)	 اللهم أرزق جميع بنات المسلمين بالزوج الصالح الذي تحبه وترضاه	 أنثرو ورود السعادة أينما ذهبتم فانية لا تملؤها بالأحزان وإياكم أن تيأسو من رحمة الله
85.  يقولون أن رمضان القادم هو الأطول نهاراً منذ ٣٣ عاماً ونحن نقول : رمضان المقبل هو الأعظم أجراً منذ ٣٣ عاماً اللهم بلغنا رمضان	86.  الشيعة النصيرية الشيطانية المجهولة	87. ولد يتيماً و مات كريماً •• ذاك سيد الخلق محمد رسول الله ♥صلو على الحبيب♥	88. إسمك في قلبي مستحيل الزمن يمحيه حتى وأنا في قبري يموت قلبي وإسمك فيه ♡ يارسول الله ♡ محمد
89.  من دسرات من لو المقابر لو أذن لكل ميت أن يتكلم لقال: أمهلتني (سوف) و (سين) سأتوب ..سوف أصدق .. سأصلي..	90. 		