LABA: Logical Layout Analysis of Book Page Images in Arabic Using Multiple Support Vector Machines

Wenda Qin
Department of Computer Science
Boston University, Boston, USA
wdqin@bu.edu

Randa Elanwar
Electronics Research Instituite, Cairo, Egypt
and Boston University, Boston, USA
relanwar@bu.edu

Margrit Betke
Department of Computer Science
Boston University, Boston, USA
betke@bu.edu

***Abstract*—Logical layout analysis, which determines the function of a document region, for example, whether it is a title, paragraph, or caption, is an indispensable part in a document understanding system. Rule-based algorithms have long been used for such systems. The datasets available have been small, and so the generalization of the performance of these systems is difficult to assess. In this paper, we present *LABA,* a supervised machine learning system based on multiple support vector machines for conducting a logical *L*ayout *A*nalysis of scanned pages of *B*ooks in *A*rabic. Our system labels the function (class) of a document(scanned book pages) region, based on its position on the page and other features. We evaluated LABA with the benchmark "BCE-Arabic-v1" dataset, which contains scanned pages of illustrated Arabic books. We obtained high recall and precision values, and found that the F-measure of LABA is higher for all classes except the "noise" class compared to a neural network method that was based on prior work.**

Keywords: Document image processing, text analysis, multi-classifier system, Arabic text documents, logical layout analysis, functional layout analysis

I. INTRODUCTION

Document analysis is widely researched for many languages, including European languages, such as English and French, and Asian languages, such as Chinese and Hindi. We here focus on the *L*ayout *A*nalysis of scanned pages of modern *B*ooks in *A*rabic, and call the proposed system *LABA*.

The motivation for developing automated methods for analyzing scanned book pages lies in improved access, because digitization enables keyword searches for book titles, authors, text and image content. Scanned documents, however, currently need the intervention of human experts to create metadata transcripts of the scanned document, which is an expensive, tedious, and slow process. Document layout analysis in conjunction with a powerful character recognition engine promises to automate this process. It can be separated into two steps: first physical and then logical layout analysis [1]. Many prior works focus on physical layout analysis, which locates the relevant homogeneous text and non-text regions in a document image. Once these have been found, logical layout analysis can start, which is the focus of our paper.

Partial funding from the National Science Foundation (1337866, 1421943) (to M.B.) and the Cairo Initiative Scholarship Program (to R.E.) is acknowledged.

Bukhari et al. [9] and Alshameri et al. [10] proposed methods for physical layout analysis of Arabic books and newspapers, respectively. The method by Bukhari et al. [9] also proposes a reading order of the contents of the analyzed pages. We are aware of only one prior work that describes a method for logical layout analysis of Arabic document book images: Hadjar and Ingold [11] proposed neural networks for physical and logical layout analyses of images of Arabic newspaper pages, called PLANET and LUNET, respectively. We reimplemented their LUNET network for logical layout analysis as a comparison system for our work. We trained it for the classes that were relevant in our dataset. We also manually provided an accurate physical layout of each document as input to the neural net to enable its best possible performance.

Inspired by the work of Le et al. [4], who used a combination of connected component features and a SVM to distinguish English text and non-text connected components, we developed a system that uses multiple SVMs to assign the connected components of an Arabic book images into different logical classes. Our logical layout analysis involves six classes. So needed to find a machine learning method for n-class classification (n=6). We adopted a one-vs.-rest classification strategy. Our system is composed of only 5 classifiers, each specialized to identify a particular logical class. Moreover, the result of one SVM can provide additional information to help another SVM to train its model and predict a result. In the end, predictions from all classifiers are combined into an n-class classification result using a voting mechanism.

Our contributions can be summarized as:
- LABA is the first to perform logical layout analysis of Arabic modern book pages,
- We worked with a public dataset, BCE-Arabic-v1 [12], to facilitate benchmarking,
- We show the superior performance of LABA compared to a reimplemented version of LUNET on this dataset.

978-1-5386-1459-4/18/$31.00 ©2018 IEEE

II. Related Works

Physical layout analysis can be performed with traditional techniques, such as the Run Length Smearing Algorithm [2], a region-based approach [3], or connected component based segmentation [4], as well as more recent deep learning techniques [5], [6]. Our focus is logical layout analysis, which is needed to recover the content structure of the scanned file and a reading sequence. It is important to note that publications related to ”logical labeling” or ”logical segmentation” typically address ”born-digital PDF documents” with hidden transcriptions in which position information of each word and image is stored [16], [17], [19], [20], [23], [24]. In digitized documents (camera-based or scanned PDF), where no hidden transcription is available, logical labeling is performed on the only source of information, the document image. The problem is challenging, particularly, since no appropriately annotated research datasets (especially in Arabic) are publicly available.

Some researchers add some sources of information to help the logical labeling process. Bloechle et al. [21], for example, proposed an interactive and dynamic learning environment to help label PDF book chapters and convert them to e-books via training of a neural net with user input. Others used OCR-processed page images as input to deliver logical labels for historical books [22].

Dengel and Shafait [18] offered a review of the state-of-the art until 2014 that describes six main approaches to logical labeling. Traditional methods (non-learning) rely on additional information, such as OCR or document domain knowledge. Learning-based methods use the raw data to analyze the document with no domain knowledge or use of heuristic rules created by experts. They also showed that the reviewed work for ”logical layout analysis” is limited and directed to labeling of business letters, invoices and periodicals and that all research datasets were private. The same comment applies to most of the research done on document analysis of technical journals, magazines. and newspapers. Tao et al. [25] also pointed out that fewer works have been published on logical labeling in the document analysis research literature compared to segmentation (physical layout analysis), due to the inherent complexity of the logical labeling problem. They also mentioned that no standardized benchmarks or evaluation sets were available despite being crucial to support research in this area.

In the Competition on Recognition of Documents with Complex Layouts [15], which was part of the International Conference on Document Analysis and Recognition (ICDAR 2015), participants put forward four methods for page layout analysis that involved logical analysis on a small part of the competition dataset. According to the competition results, described by Antonacopoulos et al. [15], the ”MHS method” maintained the highest success rate in three different scenarios among the four methods. It is based on connected component analysis and a classic layout segmentation algorithm called ”white space analysis.” There are two important steps in the MHS method: MHS first distinguishes text components from non-text components in the image, and then further classifies the two general types of components into different logical classes based on their properties such as size and position, which is similar to the idea of our proposed method.

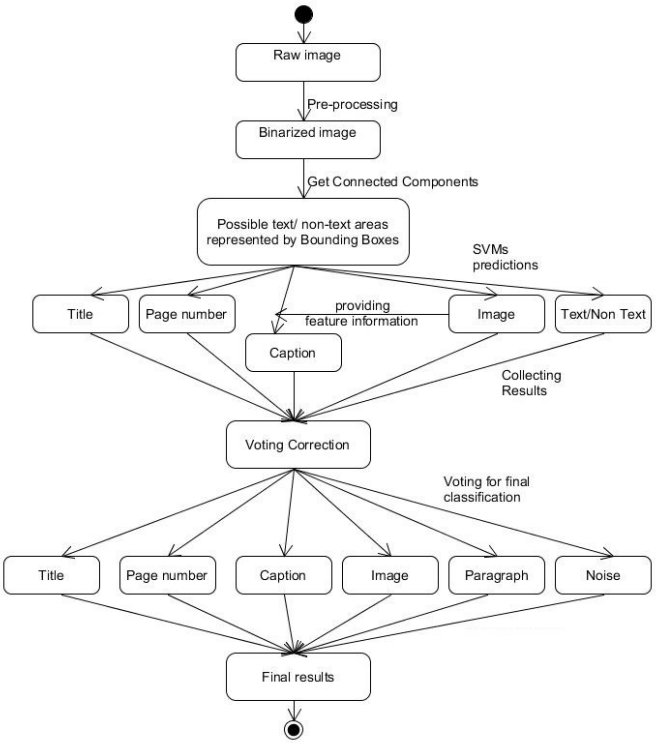

Fig. 1. Overview of proposed system LogiLAB-Arabic.

III. Method

An overview of our proposed system is shown in Fig. 1.

A. Preprocessing: Extraction of regions

The LABA system first conducts segmentation as a pre-processing step for logical layout analysis. This step extracts relevant regions from the raw document image. There are two ways of doing this: one is to use machine learning techniques to collect text and non-text area proposals. Another way is to use traditional computer vision techniques, such as X-Y cut, smearing, white-space analysis [13]. The documents of BCE-Arabic-v1 were photographed/scanned adequately. We are able to find and label most connected components in the page images using the default contour-finding function provided by OpenCV, which is based on the work by Suzuki [26].

The LABA system is based on the analysis of connected components in the document image, and it does not extract text lines or text blocks from the image. The pre-processing step of LABA is based not only on basic techniques for image pre-processing (1-3 below), but also the following observations:

- A picture is usually large, which is a feature that can be easily captured. Pictures in our dataset, however, also can contain several objects and regions with different colors or decorative patterns, which makes more difficult

to detect a picture as a single region of interest rather than detecting only irregular-shaped fragments in the picture area.
- Bounding boxes of candidate text regions are unlikely to intersect each other since there is always white space separating words. This is typically not found for bounding boxes that contain portions of pictures.
- Both text and non-text regions will not cover more than a certain percentage of the image; at the same time, their areas will not be smaller than a certain number of pixels (we use 2×2 as the smallest possible bounding box area in a document image).

According to the above observations, we set up a rule-based preprocessing algorithm as follows:
1. Convert the image into gray-scale.
2. Use Otsu's method [14] to binarize the gray-scale image.
3. Erode the image, removing regions of noise that are too small while making word contours more distinguishable.
4. Use an active contour algorithm to find contours in the processed image. Then, by using a connected component labeling algorithm, extract information about each contour.
5. By using the pixel information of each contour, transform the shape of these contours into bounding boxes. (Bounding boxes are more convenient for calculating solidity, size, and intersection information of the contour of interest without losing too much information from the original contours.)
6. Remove contours whose areas are over 60 % of the image area. Such regions are likely outside the book page (border noise). Also, remove contours whose areas are ≤ 4 pixels, which are considered to be noise.
7. Merge two bounding boxes if the center of one of them is inside the other. (After transforming contours into bounding boxes, a text bounding box can slightly overlap another text bounding box, while most of the image bounding boxes intersect with others with a large part of themselves. Thus, such merging can keep text areas close to others while merging image contours falsely separated into two separated contours.)

B. Multi-SVM Classification

Paragraph: Paragraphs make up the body of most document images. It is difficult to judge whether a text bounding box belongs to a paragraph. So conversely, if a bounding box is considered to be text, and it is not a title&header, caption, or page number, then it should be classified as a paragraph (footnote and table regions are excluded here, see Section 4). So, to determine whether a text bounding box can be considered to belong to a paragraph, our system only needs to judge whether the box contains text, and leave the rest to be decided by other classifiers.

Title&header: Titles&header are located in relatively upper positions of the document page. Their sizes should be similar to other text bounding boxes. Subtitles might appear in any non-edge area of the document page, but they are typically shorter than normal paragraph lines and may have larger sizes.

Page number: Page numbers appear as text near the edge of a page, while maintaining text bounding box sizes. They are usually independent from other parts of the text and do not have neighboring text in either a same horizontal or vertical line. Their sizes are smaller than normal text bounding boxes but much larger than salt-and-pepper noise (if any remains).

Caption: Captions can be any length and can be anywhere in the page. Sometimes a page can only consist of a single picture and its caption. Captions are difficult to distinguish from paragraphs simply based on their position and size information. However, the most significant characteristics of any caption are that they are close to and above/below a picture and they are separated from other text blocks. For the first characteristic, which is the most important one, we can first use the picture classifier to determine where the picture is in the image. Then by calculating the vertical distance between the predicted bounding box and its closest picture, we can have a direct information about the distance between the predicting area and the picture so as to determine whether it is a part of caption. For the second characteristic, although block information cannot be extracted from the image yet, for those small caption blocks which form a whole line, we can calculate the number of black pixels in the same line as a signal of a caption block. If a line contains many white pixels, then the connected components in that line are more likely to be part of a caption.

Pictures: Bounding boxes of a picture are usually much larger than any single text bounding box if the picture bounding boxes are detected and merged by the preprocessing step successfully. To distinguish large pieces of titles&header, we can additionally use the characteristic of overlapped areas (after preprocessing) mentioned in Section 2.1 to help determining the class of those bounding boxes that heavily overlap with each other as a picture. To remove noise from the remaining possible bounding boxes, our system uses their size and position information in the page to determine whether they are noise or not.

Noise: In most cases, noise appears at the edge of the image. Noise regions are typically much smaller or much larger than text regions. So if a region is considered to be non-text and it is not a part of any picture, our system leaves it labeled as noise.

Based on the above ideas, we set up several SVMs respectively. The kernel function used is "histogram intersection," which we selected due to its fast training speed. The maximum number of iterations for training is 100,000. The input features and functionalities are as follows:

1) **Text classifier:** The text classifier is used to distinguish text and non-text regions, in particular, identify pictures and noise as non-text (mainly salt-and-pepper noise). It classifies a document image based on the following features:
   a) Center.X/Image.Width
   b) Center.Y/Image.Height
   c) Width/Image.Width
   d) Height/Image.Height
   e) Number of black pixels / total number of pixels in the bounding box

TABLE I
VOTING MECHANISM. $\surd$ INDICATES A POSITIVE RESULT, $\times$ INDICATES A NEGATIVE RESULT, O INDICATES THE CLASSIFIER IS IRRELEVANT.

	Text	T & H*	Page #	Picture	Caption
Paragraph	$\surd$	$\times$	$\times$	$\times$	$\times$
T& H*	$\surd$	$\surd$	O	$\times$	O
Page #	$\surd$	O	$\surd$	$\times$	O
Picture	$\times$	$\times$	$\times$	$\surd$	$\times$
Caption	$\surd$	O	O	$\times$	$\surd$
Noise	O	O	O	O	O

*T&H stands for Title & Header

2) **Title&Header classifier:** This classifier is used to identify regions that contain titles or page headers, whose connected components are usually somewhat larger and at the upper part of the document page. The input features are: a)–d) as above and
   f) Order in Y coordinate among all bounding boxes / Number of all bounding boxes
   g) Number of black pixels when Y = Center.Y / Image.Height
3) **Page number classifier:** The page number classifier is used to identify regions that contain the document page number. It is similar to the title&header classifier, but conversely, the connected component size is much smaller and at the bottom of the page. The input features are a)–d) and f)-g)
4) **Picture classifier:**Picture classifier is used to classify picture areas in the image, they are usually much larger than other connected components. The input features area are a)–d) and
   h) Overlapped area with other bounding boxes/Image.Area,
   i) Number of bounding boxes crossing Y = Center.Y / Number of bounding boxes
5) **Caption classifier:** The caption classifier is unique. It uses the output of the picture classifier, which gives the caption classifier the position information of pictures in the page to help classify captions in the image. Connected components that are far away from the picture area will be less likely classified as captions. Thus, the input features are a), b) and
   j) Distance to the closest image bounding box in Y coordinate/Image.Height,
   k) Area of the closest image bounding box/ Image.Area,
   l) Number of black pixels when Y = Center.Y / Image.Width

By adopting the above-described multiple SVM system, we have an almost completed logical labeling result for each region-of-interest of the document image.

C. Postprocessing: Voting Mechanism

The multi-SVM system predicts labels for the input bounding boxes. To combine them into one final result, we adopt a voting mechanic to help our final system make a better decision. For example, if a bounding box is classified as non-text by the text classifier but falsely classified as a page number, which is a common situation when there is noise near the area where page number are usually located, the system can correct the mistake by using an "and" operation with both the text and page number classifiers. The voting system is described in Table I. After voting, LABA produces the class label results.

IV. EXPERIMENTS

The scanned book pages for our experiments are 200 images of the "BCE-Arabic-v1 dataset," provided by Saad et al. [12]. They are classified as "text/image," including representative areas such as pictures, paragraphs, titles or headers, page numbers, and captions. To analyze the performance of our system, we used cross validation and split the 200 images into 150 images for training and 50 images for testing. We ran this experiment four times in a round-robin manner, with different training and testing images each time, and report cumulative results. We use pixels as the basic unit that must be labeled (instead of connected components) by class membership.

LABA was implemented in C++ using OpenCV (https://opencv.org) and run on an Intel Core i7 CPU. The tuning of parameters of the SVM system is based on experimentation. To ensure that the training process finishes with high accuracy and in reasonable time, which is within 2 minutes for 150 images as training data, we set our system to force termination of the training process after 10^5 iterations. To choose the kernel function of the SVM, we compared the following types: linear, polynomial, radial basis functions, sigmoid, exponential χ^2 function, and histogram intersection kernels. Among these, the histogram intersection kernel maintained a high accuracy of classification while completing the training process within 2 minutes and was therefore chosen for the LABA system.

As a comparison system, we reimplemented the LUNET neural network for logical layout analysis described by Hadjar and Ingold [11] (reimplementation was necessary as the code was not publicly available). We trained it with our data to recognize the classes relevant in our data (note that their system recognizes additional classes that are not available in our data). In this experiment, we used the same preprocessed connected components as proposed by step 1 in our system. Additionally, we determined the regions of different logical blocks manually, so that we could calculate the features needed as input to the neural network based on a perfect segmentation.

V. RESULTS

Qualitative results for LABA and the comparison system can be seen in Figures 2 and 3. Quantitive results can be seen in Table II. We obtained high pixel class membership accuracy values of 96.5% (LABA) and 94.4% (reimplemented LUNET). For all classes, except "noise," LABA outperforms the reimplemented LUNET. The increase in performance is statistically significant in the improved precision of classifying class title&header, page number, and caption regions: 41.1 percent points in classifying page numbers, 43.1 percent points

TABLE II
PRECISION, RECALL AND F MEASURE OF THE EXPERIMENTAL RESULTS OF THE REIMPLEMENTED LUNET [11] (TOP) AND OUR LABA SYSTEM (BOTTOM)

LUNET	Precision	Recall	F measure
Noise	99.20%	31.68%	0.48
Picture	98.85%	95.76%	0.97
Paragraph	93.33%	98.03%	0.96
Title&Header	56.77%	81.75%	0.67
Page Number	58.64%	82.12%	0.68
Caption	80.90%	76.40%	0.79

LABA	Precision	Recall	F measure
Noise	49.53%	28.50%	0.36
Picture	99.46%	99.40%	0.99
Paragraph	94.44%	98.80%	0.97
Title&Header	99.98%	69.24%	0.82
Page Number	99.68%	100.00%	1.00
Caption	91.86%	77.89%	0.84

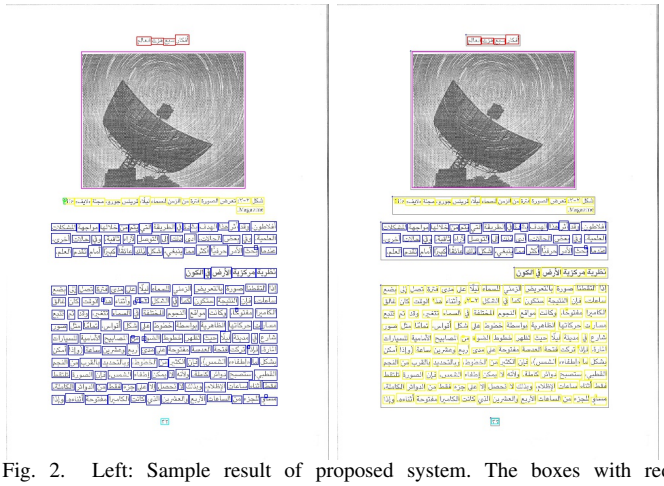

Fig. 2. Left: Sample result of proposed system. The boxes with red boundaries enclose pixels in the "title&header" class, pink boundaries pixels in the "picture" class, yellow boundaries pixels in the "caption" class, blue boundaries the "paragraph" class, cyan boundaries the "page number," and green "noise." Right: The result of the reimplemented neural network [11] on the same image. Here a paragraph was misclassified as a figure caption.

classifying the title&header region, and 11.0 percent points classifying a caption, while maintaining high precision in classifying paragraphs and pictures.

VI. DISCUSSION

The reasons why the proposed system outperforms the comparison system significantly can be understood by comparing Figures 2 and 3.

Firstly, captions of photographs in normal book pages come in various sizes. If a caption is short, its size may be similar to that of a title&header region. If the caption is long, its size is similar to that of a paragraph. Therefore, for the neural network, it is difficult to judge whether a block belongs to a caption, title&header, or paragraph in some cases. Once the block is falsely marked, every connected component inside the block will be falsely classified.

Secondly, position information is important in deciding the classes of pixels in a title&header and page number region. Especially, when the difference of the sizes of title&header and

Fig. 3. Dependence of the classification of the neural network on the precision of human annotation of the training data. If the human annotator draws the border of the page number with additional white space (top) versus snug around the characters (bottom), the network confuses the page number (cyan) for a caption (yellow).

page number regions is not large, the system will be confused if their position information is not provided. For classifying a caption, the very crucial information is the picture position in the image. Therefore, by incorporating picture position information from the picture classifier, our method has better access to classifying captions in the scanned book image.

Lastly it should be noted, the reimplemented neural network needs manually annotated training data and the decision on the location of the border of a bounding box may not always be straightforward and may differ from time to time and from person to person. Therefore, there is the risk that an area may be classified as two different classes, simply because the two corresponding marked areas have annotation differences. An example of this is shown in Figure 4. The problem may be alleviated with classical rule-based algorithms for text segmentation whose performance was discussed by Shafait et al. [13]). In LABA, the whole classification process is automated and, thus, we do not need to worry that differences in human annotation will mistakenly influence the final result.

We investigated which layouts are challenging for LABA to label correctly. When the picture is not a photograph but a diagram with text labels, as in Fig. 5, LABA does not understand that these snippets of text belong to the picture. Since LABA was not trained with such cases, due to the similar size and density of these labels to the connected components in true paragraphs, classifies them as "paragraphs."

Another problem we encountered was identifying large captions or page numbers. Connected component classification can become inaccurate when the caption contains many rows of text, making the last row far from the picture (see Figure 6), or when page numbers in test images are much larger than in training images. Distinguishing page numbers from titles becomes difficult when a page number appear at the top of the page in a test image but at the bottom of the page in the training images. Table and chart regions have not been addressed with our LABA system.

VII. CONCLUSIONS

The contribution of this paper is the logical layout labeling system LABA that can classify pictures, titles&header, paragraphs, page numbers, and captions in the scanned page of an

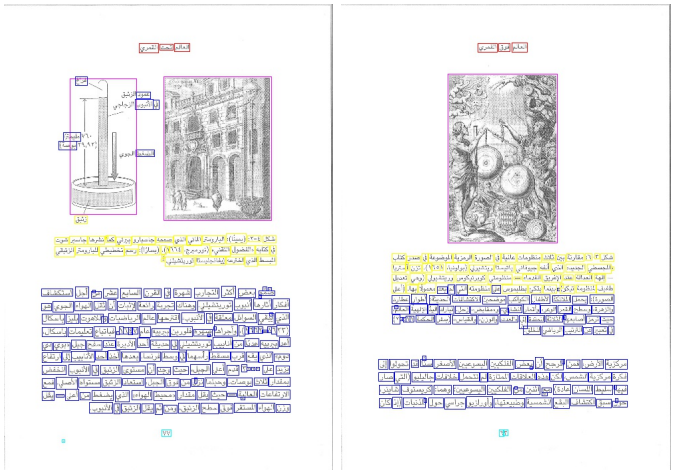

Fig. 4. Left: Text labels in a diagram were identified as "paragraph" regions by our system. The neural network was able to include such fragments since the picture boundaries were marked manually. Right: An example of a book page with a long caption that is not fully recognized by the proposed system.

Arabic book. LABA consists of a binarization and connected-components-extraction preprocessing step, a multiple-SVM prediction step, and a voting-adjustment step. LABA provides several advancements, including: 1. Providing a logical layout classifier that works robustly for Arabic book pages that have a layout commonly-occurring in the BCE-Arabic-v1 dataset (for example, page numbers are always at the bottom of the book pages, and captions are always below a picture). 2. Being able to recognize the class of a connected component instead of a whole block on a page. This provides flexibility in classifying non-square text blocks. The document image does not need to be separated into blocks prior to classification as in previous work [11]. 3. Instead of building a traditional SVM n-class classifier, we designed LABA to use the result of the picture classifier as an input to the caption classifier, which improves its performance. 4. When combining the classification results, LABA uses a voting mechanism to correct obvious mistakes made during the merging of results.

In future work, we will include additional classes into the classification system, such as diagrams and tables, and address the more complex and decorated layouts found in BCE-Arabic V2. To deal with such complex layouts, introducing methods provided by Antonacopoulos et al. [15] might be helpful. Furthermore, to understand the text content in a book image, we will also consider introducing an OCR step to our system.

We hope to facilitate the research of others by providing our code for LABA at http://www.cs.bu.edu/faculty/betke/research/LABA. Our results may be reproduced by running it on the publicly-available BCE-Arabic-v1 [12].

References

[1] Mao S, Rosenfeld A, Kanungo T. 2003. Document structure analysis algorithms: a literature survey. DRR, pp. 197–207.

[2] Wong K Y, Casey R G, Wahl F M. Document analysis system. IBM journal of research and development, 1982, 26(6): 647–656.

[3] Lin M W, Tapamo J R, Ndovie B. 2006. A texture-based method for document segmentation and classification. South African Computer Journal, 36: 49–56.

[4] Le V P, Nayef N, Visani M, et al. 2015. Text and non-text segmentation based on connected component features. 13th International Conference on Document Analysis and Recognition, pp. 1096–1100.

[5] Moysset B, Kermorvant C, Wolf C, et al. 2015. Paragraph text segmentation into lines with recurrent neural networks. In 13th International Conference on Document Analysis and Recognition, pp. 456–460.

[6] Wang L, Fan W, Sun J, et al. 2015. Text line extraction in document images. 2015 13th International Conference on Document Analysis and Recognition, pp. 191–195.

[7] Corbelli A, Baraldi L, Balducci F, et al. 2016. Layout analysis and content classification in digitized books. In Italian Research Conference on Digital Libraries, pp. 153–165.

[8] Tao X, Tang Z, Xu C. 2014. Contextual modeling for logical labeling of PDF documents. Computers & Electrical Engineering, 40(4): 1363–1375.

[9] Bukhari S S, Shafait F, Breuel T M. 2011. High performance layout analysis of Arabic and Urdu document images, In International Conference on Document Analysis and Recognition. pp. 1275–1279.

[10] Alshameri A, Abdou S, Mostafa K. 2012. A combined algorithm for layout analysis of Arabic document images and text lines extraction. International Journal of Computer Applications, 49(23):30–37.

[11] Hadjar K, Ingold R. 2005. Logical labeling of Arabic newspapers using artificial neural nets. In the Eighth International Conference on Document Analysis and Recognition, pp. 426–430.

[12] Saad R S M, Elanwar R I, Kader N S, et al. 2016. BCE-Arabic-v1 dataset: Towards interpreting Arabic document images for people with visual impairments. In the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments, 5 pp.

[13] Shafait F, Keysers D, Breuel T. 2006. Performance comparison of six algorithms for page segmentation. Document Analysis Systems VII: 368–379.

[14] Otsu N. 1979. A threshold selection method from gray-level histograms. IEEE Trans. on Systems, Man, and Cybernetics 9(1): 62–66.

[15] Antonacopoulos, A., Clausner, C., Papadopoulos, C. and Pletschacher, S. 2015. Competition on Recognition of Documents with Complex Layouts. In 13th IEEE International Conference on Document Analysis and Recognition, pp. 1151–1155.

[16] Adrian, W.T., Leone, N., Manna, M. and Marte, C., 2017. Document Layout Analysis for Semantic Information Extraction. In Conference of the Italian Association for Artificial Intelligence, pp. 269–281.

[17] Rahman, M.M. and Finin, T., 2017. Deep Understanding of a Documents Structure. In 4th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies.

[18] Dengel, A. and Shafait, F. 2014. Analysis of the Logical Layout of Documents, D. Doermann, K. Tombre (eds.), Handbook of Document Image Processing and Recognition, Springer-Verlag, pp. 177–222

[19] Tkaczyk, D., Szostek, P., Dendek, P.J., Fedoryszak, M. and Bolikowski, L., 2014, April. Cermine–automatic extraction of metadata and references from scientific literature. In 11th IAPR International Workshop on Document Analysis Systems, pp. 217–221.

[20] Hamza, H., Belad Y., Belad, A., and Chaudhuri, B.B. An end-to-end administrative document analysis system. 2008. The Eighth IAPR International Workshop on Document Analysis Systems, pp.175–182.

[21] Bloechle, J.L., Rigamonti, M. and Ingold, R., 2012. OCD Dolores-recovering logical structures for dummies. In 10th IAPR International Workshop on Document Analysis Systems, pp. 245–249.

[22] Gander, L., Lezuo, C. and Unterweger, R., 2011. Rule based document understanding of historical books using a hybrid fuzzy classification system. In Workshop on Historical Document Imaging and Processing, pp. 91–97.

[23] Tuarob, S., Mitra, P. and Giles, C.L., 2015. A hybrid approach to discover semantic hierarchical sections in scholarly documents. In 13th International Conference on Document Analysis and Recognition, pp. 1081–1085.

[24] Gao, L., Zhong, Y., Tang, Y., Tang, Z., Lin, X. and Hu, X., 2011. Metadata Extraction System for Chinese Books. In International Conference on Document Analysis and Recognition, pp. 749–753).

[25] Tao, X., Tang, Z., Xu, C. and Wang, Y., 2014, April. Logical labeling of fixed layout PDF documents using multiple contexts. In 11th IAPR International Workshop on Document Analysis Systems, pp.360–364.

[26] Suzuki, S. 1985 "Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing, 30(10): 32–46.