Article

JIS

Journal of Information Science
2022, Vol. 48(2) 268–279

DOI: 10.1177/0165551520961256
journals.sagepub.com/home/jis
SAGE

Text and metadata extraction from scanned Arabic documents using support vector machines

Wenda Qin
Boston University, USA

Randa Elanwar
Electronics Research Institute, Egypt

Margrit Betke
Boston University, USA

Abstract

Text information in scanned documents becomes accessible only when extracted and interpreted by a text recognizer. For a recognizer to work successfully, it must have detailed location information about the regions of the document images that it is asked to analyse. It will need focus on page regions with text skipping non-text regions that include illustrations or photographs. However, text recognizers do not work as logical analyzers. Logical layout analysis automatically determines the function of a document text region, that is, it labels each region as a title, paragraph, or caption, and so on, and thus is an essential part of a document understanding system. In the past, rule-based algorithms have been used to conduct logical layout analysis, using limited size data sets. We here instead focus on supervised learning methods for logical layout analysis. We describe LABA, a system based on multiple support vector machines to perform logical *L*ayout *A*nalysis of scanned *B*ooks pages in *A*rabic. The system detects the function of a text region based on the analysis of various images features and a voting mechanism. For a baseline comparison, we implemented an older but state-of-the-art neural network method. We evaluated LABA using a data set of scanned pages from illustrated Arabic books and obtained high recall and precision values. We also found that the F-measure of LABA is higher for five of the tested six classes compared to the state-of-the-art method.

Keywords

Arabic text documents; document image processing; layout analysis; metadata extraction; multi-classifier system

1. Introduction

One of the most important functionalities of digital libraries, which is highly appreciated by users, is the ability for advanced search. 'Advanced search' means that the user can specify which part of a document should be searched – it may be a search for an author, a book title, a keyword in an abstract and so on. Queries can be accessed easier and processed faster if the document is digital, also called 'born digital', rather than digitised, that is, a scanned image of a document. Scanned documents currently need the intervention of human experts to create a transcript that includes the important metadata, a process that is expensive, tedious and slow.

Research about logical layout analysis promises to provide a solution for the costly annotation process by detecting the functionality of the various parts of a document layout automatically (see an example in Figure 1). In conjunction with a powerful text recognition engine, logical layout analysis can then provide the meaning of text images in a natural

Corresponding author:
Randa Elanwar, Electronics Research Institute, Joseph Tito St, Huckstep, El Nozha, Cairo 11843, Egypt.
Email: randa.elanwar@eri.sci.eg

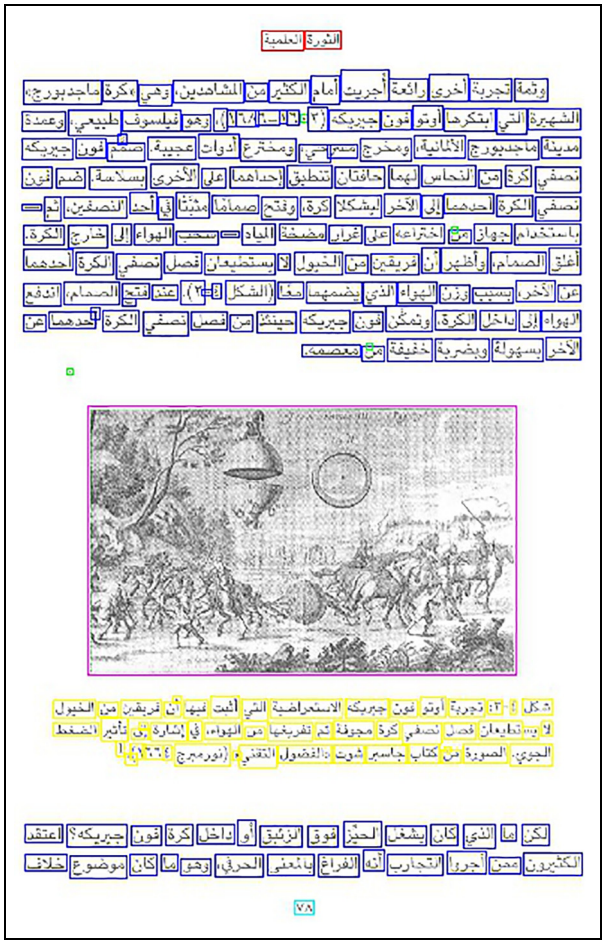

Figure 1. An Arabic book page with regions and their functions, as detected by LABA. The regions with red boundaries enclose pixels in the 'Title & Header' class.
Pink means the 'Picture' class, yellow the 'Caption' class, blue the 'Paragraph' class, cyan the 'Page number' class, and green the 'Noise' class.

language and formatting highlights. Finding a method to reveal the reading order of text paragraphs is one of many additional concerns of logical analysis research. According to Dengel and Shafait [1], document logical layout analysis provides other higher-level functionality options like automatic routing of business letters, automatic processing of invoices and within-book navigation facility.

Document image understanding has two main components with respect to the analysis of the document layout, called 'physical' and 'logical' layout analyses [2]. Many prior works focus on physical layout analysis, which locates the relevant homogeneous regions in a document image and can be done with traditional techniques, such as the Run Length Smearing Algorithm [3], a region-based approach [4], or connected component (CC)-based segmentation [5], as well as more recent deep learning techniques [6,7]. Once text and non-text areas in the document have been found, logical layout analysis can start, which is the focus of our article.

The literature is rich with publications dedicated to physical and logical analysis of European languages, particularly English, but also French, Greek, German and Latin, as well as Asian languages like Chinese or Hindi. Corbelli et al. [8], for example, proposed a support vector machine (SVM)-based method for logical segmentation of English document images. Publications for Arabic document analysis, however, are very limited. Among these, publications dedicated to methods that analyse modern documents like books and newspapers are even rarer. The limited literature has concentrated on analysing historical manuscripts and newspapers [9,10], which have layouts that are significantly different from the layout of modern books that we consider in this article. We here focus on LABA, the *L*ayout *A*nalysis of scanned

pages of modern *B*ooks in *A*rabic. We presented our work on LABA to a limited audience at a workshop [11] and here describe our system in detail to a broader journal audience.

For Arabic document analysis, Hadjar and Ingold [10] proposed neural networks for physical and logical layout analyses of images of newspaper pages, called PLANET and LUNET, respectively. We reimplemented the LUNET network for logical layout analysis as a baseline comparison system for our work. We trained it for the layout classes that were relevant in our data set. We also manually provided an accurate physical layout of each document as input to the neural net to enable its best possible performance.

Most logical analysis algorithms in the literature use heuristic rules based on geometric features of page image regions or smaller components to identify the corresponding classes. Progress was made with a learning-based approach, proposed by Le et al., [5] who used a combination of CC features and an SVM classifier to distinguish English text and non-text CCs. We also follow the machine-learning route, but developed a system that instead uses *multiple* SVMs to assign the CCs of an Arabic book images into different logical classes.

Our logical layout analysis involves six classes: paragraphs, titles & headers, page numbers, captions, pictures and noise regions. Our task is therefore a multi-class machine learning problem that we solve with a classification strategy that compares each class against the remaining classes. Our system LABA is therefore composed of multiple classifiers, each specialised 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, $n = 6$, using a voting mechanism.

The contributions of our work are as follows:

- We present LABA, a logical layout analysis system that determines the functionality of images regions in scanned pages of modern Arabic books.
- Experiments show that LABA outperforms the baseline system, which is a reimplemented version of previous work [10] on our data set.

2. Related works

As was explained by Adrian et al. [12]:

> [The] PDF format [of a document] does not guarantee a machine-readable representation of [its] layout and structure. Instead, [layout and structure] must often be 'recovered'.

Logical labelling is an essential procedure to recover PDF content structure and the reading sequence of text regions. It is important to note that other synonyms of 'logical analysis' like 'information/metadata extraction', 'logical segmentation', 'semantic segmentation' and 'logical labeling' do not have exactly the same meaning. Most of the publications related to the latter three terms address born digital PDF documents with hidden transcriptions that store the position information of each word and use ontology/semantic information, that is, based on the meaning of data to assist the geometric features stored in the hidden and perform several tasks like effective reading, document understanding, file/information retrieval, document clustering/classification and so on [12–18]. Among these approaches, we highlight the work by Tao et al. [19], who focused on English and Chinese documents in born digital PDF images, and by Yang et al. [18], who proposed a deep encoder–decoder model for semantic segmentation of English-born digital documents. To handle the need of a large training data set, the authors synthesised 135,000 English documents.

Logical labelling of digitised documents, that is, camera-based or scanned PDFs, is more challenging than logical labelling of born digital files because hidden transcriptions are not available. Logical labelling must be performed on the only source of information available, the document image and the geometric features of its content. Moreover, the problem is challenging, since appropriately annotated research data sets, especially in Arabic, are not publicly available. Appropriately synthesising scanned documents is also more difficult.

Some researchers tried to add some sources of information to help logical labelling – Bloechle et al. [20], for example, used an interactive and dynamic learning environment to help label PDF book chapters and convert them to reflowable e-books by training multilayer perceptron (MLP) networks and enabling user interaction via the graphical user interface. Others used optical character recognition (OCR)-processed page images as input to deliver logical labels for historical books via heuristic rules [21].

Dengel and Shafait [1] offered a review of the state of the art, which included six main approaches for logical labelling. Many of them require the existence of additional information like OCR results or document domain knowledge, for example, knowledge about the layout of business letters or invoices. An example of such an approach is the work by

Meier et al. [22], which analyzes OCR-preprocessed images of newspaper layouts with a fully convolutional network. In contrast, other learning-based methods use the raw physical data to analyse the document without prior knowledge or expert-established heuristic rules, simply letting the system learn the labelling function by itself. Dengel and Shafait [1] also showed that the reviewed work for 'logical layout analysis' was limited and directed on understanding business letters, invoices and periodicals.

In addition, Dengel and Shafait [1] pointed out that understanding books was researched in a different way, by analysing page sections and generating a table of contents to make the digitised books searchable. They also found that all the research data sets they were aware of were not publicly available. The same comment applies for most of the research done on layout analysis of technical journals, magazines and news papers. Similar remarks were made by Tao et al. [23], who noted that fewer publications are available in the logical labelling literature compared to the segmentation literature (i.e. the literature on physical layout analysis) due to the inherent complexity of the task of automated logical layout analysis. Tao et al. [23] regretted that no standardised benchmarks or evaluation sets were available despite being crucial in moving the research field forward.

In the Competition on Recognition of Documents with Complex Layouts [24], which was part of the International Conference on Document Analysis and Recognition (ICDAR) in 2015, participants put forward four methods for page layout analysis that involved logical analysis on a small part of the competition data set. According to the competition results, described by Antonacopoulos et al. [24], the 'MHS method' maintained the highest success rate in three different scenarios among the four methods. It is based on CC 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. The two-stage approach has similarities with our idea to combine classifier results. We were also inspired by the work of Le et al. [5], who used an SVM to distinguish the CCs of English text and non-text regions. We, however, instead developed a system that uses *multiple* SVMs to distinguish *Arabic* text from non-text CCs and determine the logical layout of the document.

3. Method

LABA does not have distinct physical (text vs non-text) versus logical (logical text subclasses) layout stages. Instead, the classification decision in the physical analysis is delayed and combined with logical classification decisions. In this way, LABA can classify six types of regions in the layout that contain the classes *paragraph, title, page number, caption, image* and *noise regions*.

Instead of performing direct n-class classification, $n = 6$, LABA reduces confusion by gathering labels with highest similarity in one class making a soft initial decision until more information from the other SVM outputs to help resolve the confusion. Accordingly, LABA is composed of separate SVM classifiers each is trained to identify a particular logical class, using a one-compared-to-the-rest approach. Predictions from all classifiers are combined through voting to generate final decisions with regards to the six classes.

LABA is trained and tested using BCE-Arabic v1, the first data set that was made publicly available for the purpose of layout analysis research. For evaluation and comparison, we reimplemented the LUNET network by Hadjar and Ingold [10] to perform logical layout analysis. LUNET is an interactive logical labelling system based on MLP networks, trained on a limited private data set created from three newspapers. In particular, an MLP neural network was trained for each different newspaper layout because, at the time, the authors found it difficult to create a general network for all document classes.

To be able to perform comparison with LUNET, we used the same training data as LABA. We re-trained the networks to our class types, using corresponding accurate, manually segmented input components to insure high performance of the neural nets.

An overview of the LABA system architecture is shown in Figure 2. The subsections below explain the LABA system stages in detail.

3.1. LABA's extraction of document regions

Generally, logical layout systems first conduct segmentation as a first step. This step extracts relevant regions from the raw document image, either using machine learning techniques to collect text and non-text area proposals, or using traditional computer vision techniques, such as X-Y cut, smearing, white-space analysis [25]. However, LABA is based on

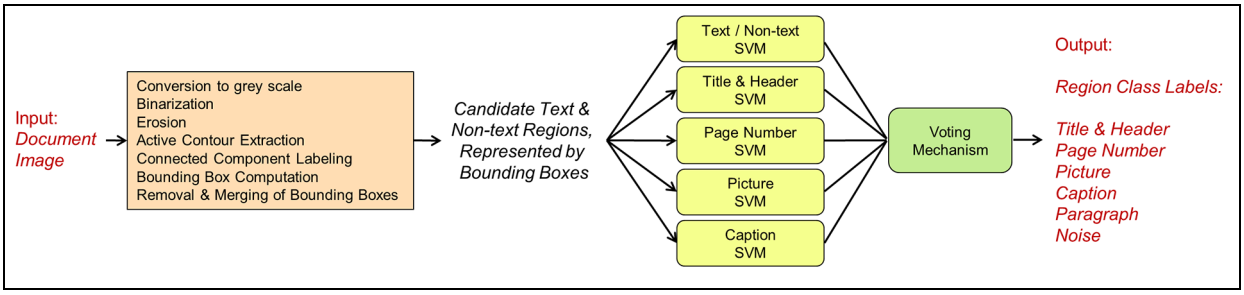

Figure 2. Overview of the LABA system: extraction of candidate regions (orange) is followed by the processing a collection of five SVMs (yellow), and finally a voting mechanism (green) that uses the SVM outputs to finalise the region class labels.

the analysis of CCs in the document image, and it does not extract text lines or text blocks from the image. LABA maintains a bounding box representation of the CCs.

The documents of the BCE-Arabic-v1 data set were scanned adequately. We are able to find and label most CCs in the page images using the default contour-finding function provided by OpenCV, which is based on the work by Suzuki [26].

The following observations about our data set help us in designing the region-extraction approach of LABA:

- A picture usually takes up a considerable portion of a book page. This is a property that can be analysed relatively easily. Because the pictures in our Arabic book page data set, however, also can contain several objects and decorative patterns, it is challenging to detect a picture as a single region of interest. Instead, irregular-shaped fragments of the picture area may be found.
- Regions that contain falsely separated portions of pictures typically intersect each other by a significant amount, and so merging of their bounding boxes is beneficial. If two regions have a large intersection, the centres of their bounding boxes are contained inside this intersection region.
- Candidate text regions are less likely to intersect each other, since words are separated by background pixels (white space). Merging of bounding boxes of text regions with slight intersections may not be beneficial because this operation may combine text regions that should stay separate, for example, a word in a paragraph and a page number that appears right below the word.
- Non-background text and non-text regions will not cover more than a certain percentage of the image, and, therefore, extremely large candidate regions have likely been falsely segmented (e.g. include border noise due to poor document scanning) and can be discarded.
- Non-background regions of interest cannot be extremely small. Small regions likely contain 'salt-and-pepper' scanning noise. Therefore, bounding boxes with fewer than 2×2 pixel can be discarded.

Considering the above observations, we designed the region-extraction algorithm (REA) shown as Algorithm 1.

Algorithm 1.
Region-Extraction Algorithm (REA).
Input: Document Image
1. Convert the image into grey-scale.
2. Binarize the grey-scale image using Otsu's method [27].
3. Erode the image and thus make word contours more distinguishable and remove small regions of noise.
4. Apply an active contour algorithm to find contours, that is., connected non-background pixels, in the processed image.
5. Use a connected component labelling algorithm to extract information about each contour.
6. Compute a bounding box of each contour.
7. Remove contours whose areas are over 60 percent of the image area.
8. Remove contours whose areas are lower or equal than 4 pixels.
9. Merge two bounding boxes if the centre of one of them is inside the other one.

Output: Extracted Binarized Regions in Bounding Boxes

3.2. The six classes of document regions

We define six classes of document regions based on their functionality in the document – titles, page numbers, captions, pictures, paragraphs and noise. We now define each class and discuss their unique properties, which will be leveraged in the classification step of LABA.

3.2.1. Title & Header. Titles and headers are located in relatively upper positions of the document page. The sizes of text boxes within a title or header are typically larger than text bounding boxes in other regions of the document. 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.

3.2.2. Page number. Page numbers appear as text near the edge of a page and occur in sizes similar or smaller than text bounding box sizes. They are usually placed away from other parts of the text and do not have neighbouring 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).

3.2.3. 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 distinctive properties of captions are that they are close to and above/below a picture and they are separated from other text blocks. For the first property, which is the most important one, LABA 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, LABA can obtain 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 property, although block information cannot be extracted from the image yet, for those small caption blocks which form a whole line, LABA 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 CCs in that line are more likely to be part of a caption.

3.2.4. Pictures/images. Bounding boxes of a picture are usually much larger than any single-text bounding box if the picture bounding boxes are detected and merged successfully by LABA's REA. To distinguish large pieces of titles or headers, we can additionally use the size of the intersection of the regions to help determine the class. Those bounding boxes that heavily overlap with each other are typically picture segments. To remove noise from the remaining possible bounding boxes, LABA uses their size and position information in the page to determine whether they are noise or not.

3.2.5. Paragraph. Paragraphs make up the body of most document images and contain numerous individual text boxes, one box per word, separated by small regions of background pixels (white space). LABA considers every text box part of a paragraph if the text box is not part of a title or header, caption or page number. So, to determine whether a text bounding box can be considered to belong to a paragraph, LABA only needs to judge whether the box contains text and leave the rest to be decided by other classifiers.

3.2.6. 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, LABA labels it as noise.

3.3. LABA's multiple SVM classifiers

We base our design of LABA's multi-class classification step on the properties of document regions described above. LABA uses five SVMs. For each SVM, the kernel function used is 'histogram intersection', which we selected due to its fast training speed.

The input features and functionalities of each of the five SVMs 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 region in a document image based on the following features:

(a) Region.Centre.X/Image. Width.
(b) Region.Centre.Y/Image. Height.
(c) Region. Width/Image. Width.
(d) Region. Height/Image. Height.
(e) Number of black pixels/total number of pixels in the bounding box.

2. Title & Header classifier: This classifier is used to identify regions that contain titles or page headers, whose CC bounding boxes 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 = Region.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 CC size is much smaller and at the bottom of the page. The input features are (a)–(d) and (f)–(g).
4. Picture classifier: The picture classifier is used to classify picture areas in the image, which are usually much larger than other CCs. The input features area are (a)–(d) and
   (h) Overlapped area with other bounding boxes/Image. Area,
   (i) Number of bounding boxes crossing Y = Region.Centre.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. CCs 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 = Region. Centre.Y/Image. Width.

The five-class SVM system is visualised in Figure 2. Each SVM computes a binary output which is then passed into the voting step of LABA, as described in the next section.

3.4. LABA's voting mechanism

LABA uses a voting mechanism to improve upon any single SVM output by combining multiple 'votes' according to the rules shown in Table 1. For a region to obtain the final label 'Paragraph', for example, the text classifier should indicate a positive vote and the four other SVMs negative votes. If a region is falsely classified as non-text by the text classifier but correctly classified as a 'Page number' region (a common situation when there is noise near the location of the page number) by the page number SVM and also correctly classified as not being a 'Picture' by the picture SVM, the voting mechanism of LABA decides that the region indeed contains a page number, thus correcting the mistake the text SVM made.

4. Experiments

4.1. Data set

A set of 200 scanned book pages from the publicly available BCE-Arabic-v1 data set [28] was selected from the 'text/image' document category, which contains pages that may include one or more of the following classes: pictures,

Table 1. LABA's voting mechanism

		SVM classifiers				
		Text	Title & Header	Page number	Picture	Caption
Class of region	Paragraph	√	×	×	×	×
	Title & Header	√	√	O	×	O
	Page number	√	O	√	×	O
	Picture	×	×	×	√	×
	Caption	√	O	O	×	√
	Noise	×	O	O	×	O

Each table row shows which classifier is relevant for predicting the class of the region and whether the region is determined by a positive or negative classifier result, that is, vote. The symbol √ indicates a positive vote, the symbol × indicates a negative vote, and the symbol O indicates that the classifier is irrelevant.

paragraphs, titles or headers, page numbers, and captions. We re-annotated the data using pixels as the basic unit (instead of CCs), labelling each pixel by its region class membership.

4.1. Training. 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 was based on experimentation. To ensure that the training process finishes with high accuracy and in reasonable time, which is within 2 min 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 min and was therefore chosen for the LABA system.

We set up the five SVMs with the 'histogram intersection' kernel function, selected for its fast training ability. Training the SVMs takes 3–4 min. The maximum number of iterations for training is 100,000.

4.2. Testing methodology. To analyse the performance of our LABA 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 labelled (instead of CCs) by class membership.

4.3. Baseline comparison. As mentioned above, to be able to compare our system performance to other researchers' work, we reimplemented the LUNET neural network for logical layout analysis described by Hadjar and Ingold [10]. Reimplementation was necessary as the code was not publicly available. LUNET recognises additional classes that are not available in our data. We trained our LUNET version with our data to recognise the classes relevant in our data. In the comparison experiment, we used the same CCs as computed by the REA, the first step of our LABA system. In addition, we determined the regions of different logical blocks manually, so that LUNET could calculate the features needed as input to the neural network based on a perfect segmentation.

5. Results

Quantitative results of LABA and LUNET for each of the six classes are reported in Table 2.

We obtained high pixel class membership accuracy values of 96.5% (LABA) and 94.4% (reimplemented LUNET). For all classes, except the 'Noise' class, LABA outperforms the reimplemented LUNET in the F measure. For the precision measure, the difference in performance is statistically significant for estimation of the 'Title & Header' class (43 percent points), the 'Page number' class (41 percent points), and the 'Caption' class (12 percent points).

Table 2. Precision, recall and F measure of the experimental results of the reimplemented LUNET (top) and our LABA system (bottom)

LUNET	Precision	Recall	F measure
Noise	**0.99**	**0.32**	**0.48**
Picture	0.99	0.96	0.97
Paragraph	0.93	0.98	0.96
Title & Header	0.57	**0.82**	0.67
Page number	0.59	0.82	0.68
Caption	0.80	0.76	0.79
LABA	**Precision**	**Recall**	**F measure**
Noise	0.50	0.289	0.36
Picture	**1.00**	**0.99**	**0.99**
Paragraph	**0.94**	**0.99**	**0.97**
Title & Header	**1.00**	0.69	**0.82**
Page number	**1.00**	**1.00**	**1.00**
Caption	**0.92**	**0.78**	**0.84**

Note: The table compares two systems LUNET and LABA. For each element of comparison (Noise, Picture, etc.) the system with the higher merits is shown in bold face.

Figure 3. Qualitative Results. Left: Result by LABA. 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' class and green the 'Noise' class. Right: The result of the reimplemented LUNET. Here a paragraph was misclassified as a figure caption.

We also provide qualitative results: An example for which LABA outperforms the reimplemented baseline system LUNET can be seen in Figure 3. An example of a misclassication of diagram text as belonging to the 'Paragraph' class by LABA is shown in Figure 4. We discuss reasons for system performance differences and limitations in the next section.

6. Discussion

The reasons why the proposed system outperforms the baseline comparison system significantly can be understood by investigating Figure 3.

First, 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. If the caption is long, its size is similar to that of a paragraph. Therefore, for the LUNET neural network, it can be difficult to judge whether a text region belongs to the classes 'Paragraph', 'Caption' or 'Title & Header'. Once the text region is falsely marked, every CC inside the bounding box will be falsely classified.

Second, 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 the 'Title & Header' and '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.

Finally, 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 5. The problem may be alleviated with classical rule-based algorithms for text segmentation whose performance was discussed by Shafait et al. [25]. In LABA, the whole classification process is automated and, as a result, we do not need to worry that differences in human annotation will mistakenly influence the final result.

We investigated which layouts of book pages are challenging for our system to label correctly. We noted that when the picture is not a photograph but a diagram with text labels (see Figure 4), our system does not understand that these single words belong to the picture. Since our system was not trained with such cases, due to the similar size and density of these labels to the CCs in true paragraphs, our system classified them as 'Paragraph' text regions.

Another problem we encountered was identifying long captions or large-size page numbers. CC classification can become inaccurate when the caption contains many rows of text, making the last row far from the picture or when page numbers are of large size; these cases are mistakenly classified as 'Paragraph'. Distinguishing page numbers from titles becomes difficult when a page number appears at the top of the page in a test image not at the bottom (like the training

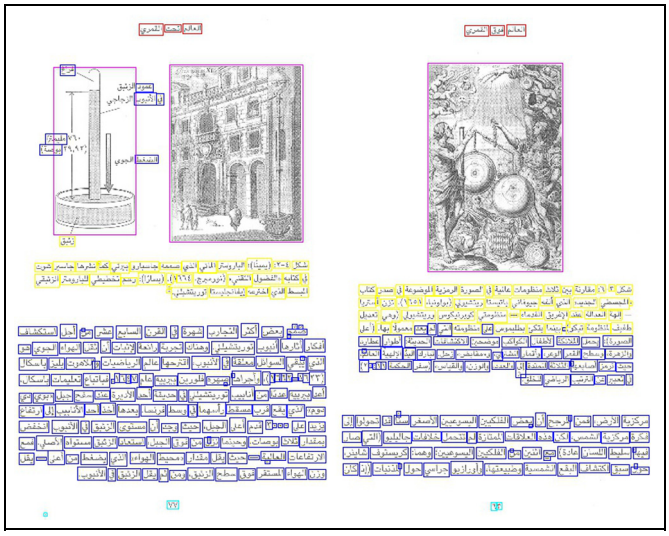

Figure 4. Left: Text labels in a diagram were identified as 'Paragraph' regions by LABA. The LUNET was able to include such fragments into the diagram region since the picture boundaries were marked manually. Right: An example of a book page with a very long caption that is not fully recognised by LABA.

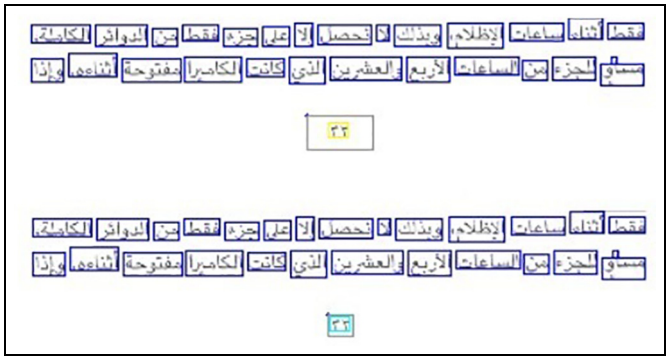

Figure 5. Dependence of the classification of the neural network on the precision of human annotation of the training data. If the human annotator draws the border of the page number with additional white space (top) versus snug around the characters (bottom), the network confuses the page number (cyan) for a caption (yellow).

examples). Table and chart regions are very challenging, since they share common features with both text and images. These are not handled by our current LABA system but will be addressed in future work.

Future work will also ensure that the data set includes other challenges such as paragraphs with widow and orphan lines, which are text lines separated from their paragraph at the top or bottom of a page, and pages with extreme scanning noise. Note that the BCE-Arabic-v1 data set was produced by scanning and digitising Arabic books in a university library [28], a process that did not result in extremely noisy documents.

As explained in the introduction, a motivation for conducting research on logical layout analysis is the need for advanced search capabilities in digitised documents. In future work, we will include LABA in a more general system that enables advanced search functionalities for digitised Arabic documents. Given the small size of Arabic data sets available for logical layout analysis research, we suggest that the proposed SVM-based approach is currently more prudent than the use of a deep neural network. A multi-class deep model would likely overfit to the limited training data. However, deep learning will play a role in our future work: We will investigate whether the use of two encoder–decoder deep networks, one for the visual information, and one for the natural language representation, can provide feature embeddings that can be used as inputs to the multi-SVM system. In particular, we will work with a publicly available deep model that can provide natural language embeddings in multiple languages [29].

7. Conclusion

The contribution of this article is a logical layout labelling system that can classify pictures, titles, paragraphs, page numbers and captions in the scanned pages of Arabic books. The proposed LABA system consists of a region-extraction step, a multiple-SVM prediction step, and a voting-adjustment step. LABA provides several advancements: LABA is a logical layout classification system that works robustly on Arabic book pages with layouts that commonly occur in the BCE-Arabic-v1 data set. LABA works on a connected component level rather than a block or region level. This provides the flexibility to classify arbitrarily shaped text blocks and non-Manhattan layouts. In LABA, the document image does not need to be separated into blocks prior to classification as in previous work. LABA performs classification using the binary votes of six SVM classifiers that complement each other and makes a final six-class prediction based on interpreting combinations of these votes. LABA performed accurately on its evaluation set BCE-Arabic-v1 and outperformed a baseline model significantly.

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. Furthermore, to understand the text content in a digitised book page, we will also consider introducing an OCR step to our system and use a deep model to compute visual and textual embeddings.

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 this code on the publicly available BCE-Arabic-v1 [28].

Acknowledgements

The authors acknowledge partial funding from the National Science Foundation (1337866, 1421943) (to M.B.) and the Cairo Initiative Scholarship Program (to R.E.).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

ORCID iD


Randa Elanwar https://orcid.org/0000-0003-3690-7141

References

[1] Dengel A and Shafait F. Analysis of the logical layout of documents. In: Doermann D and Tombre K (eds) *Handbook of document image processing and recognition*. London: Springer, 2014, pp. 177–222.
[2] Mao S, Rosenfeld A and Kanungo T. Document structure analysis algorithms: a literature survey. *Int Soc Optic Photon* 2003; 5010: 197–207.
[3] Wong KY, Casey RG and Wahl FM. Document analysis system. *IBM J Res Develop* 1982; 26(6): 647–656.
[4] Lin MW, Tapamo JR and Ndovie B. A texture-based method for document segmentation and classification. *South African Comput J* 2006; 36: 49–56.
[5] Le VP, Nayef N, Visani M et al. Text and non-text segmentation based on connected component features. In: *2015 13th international conference on document analysis and recognition (ICDAR)*, Tunis, Tunisia. 23–26 August 2015, pp. 1096–1100. New York: IEEE.
[6] Moysset B, Kermorvant C, Wolf C et al. Paragraph text segmentation into lines with recurrent neural networks. In: *2015 13th international conference on document analysis and recognition (ICDAR)*, Tunis, Tunisia. 23–26 August 2015, pp. New York: 456–460. IEEE.
[7] Wang L, Fan W, Sun J et al. Text line extraction in document images. In: In: *2015 13th international conference on document analysis and recognition (ICDAR)*, Tunis, Tunisia. 23–26 August 2015, pp. 191–195. New York: IEEE.
[8] Corbelli A, Baraldi L, Balducci F et al. Layout analysis and content classification in digitized books. In: *Italian research conference on digital libraries*, Florence, 4–5 February 2016, pp. 153–165. Cham: Springer.
[9] Almutairi A and Almashan M. Instance segmentation of newspaper elements using mask R-CNN. In: *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, Boca Raton, FL, 16–19 December 2019, pp. 1371–1375. New York: IEEE.
[10] Hadjar K and Ingold R. Logical labeling of Arabic newspapers using artificial neural nets. In: *Eighth international conference on document analysis and recognition (ICDAR'05)*, Seoul, South Korea, 31 August–1 September 2005, pp. 426–430. New York: IEEE.

[11] Qin W, Elanwar R, Betke M. LABA: Logical Layout Analysis of Book Page Images in Arabic Using Multiple Support Vector Machines. In: *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, 2018 Mar12, pp. 35–40. IEEE.
[12] Adrian WT, Leone N, Manna M et al. Document layout analysis for semantic information extraction. In: *Conference of the Italian association for artificial intelligence*, Bari, 14–17 November 2017, pp. 269–281. Cham: Springer.
[13] Hamza H, Belaïd Y, Belaïd A et al. An end-to-end administrative document analysis system. In: *2008 the eighth IAPR international workshop on document analysis systems*, Nara, Japan, 16–19 September 2008, pp. 175–182. New York: IEEE.
[14] Rahman MM and Finin T. Deep understanding of a documents structure. In: *2017 4th IEEE/ACM international conference on big data computing, applications and technologies*, 5–8 December 2017, pp. 63–73. New York: IEEE.
[15] Tkaczyk D, Szostek P, Dendek PJ et al. Cermine–automatic extraction of metadata and references from scientific literature. In: *2014 11th IAPR international workshop on document analysis systems*, New York, 7–10 April 2014, pp. 217–221. New York: IEEE.
[16] Tuarob S, Mitra P and Giles CL. A hybrid approach to discover semantic hierarchical sections in scholarly documents. In: *13th international conference on document analysis and recognition (ICDAR)*, Tunis, Tunisia, 23–26 August 2015, pp. 1081–1085. New York: IEEE.
[17] Gao L, Zhong Y, Tang Y et al. Metadata extraction system for Chinese books. In: *2011 international conference on document analysis and recognition*, Beijing, China, 18–21 September 2011, pp. 749–753. New York: IEEE.
[18] Yang X, Yumer E, Asente P et al. Learning to extract semantic structure from documents using Multimodal fully Convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, 21–26 July 2017, pp. 5315–5324. New York: IEEE.
[19] Tao X, Tang Z and Xu C. Contextual modeling for logical labeling of PDF documents. *Comput Elect Eng* 2014; 40(4): 1363–1375.
[20] Bloechle JL, Rigamonti M and Ingold R. OCD Dolores-recovering logical structures for dummies. In: *2012 10th IAPR international workshop on document analysis systems*, Gold Cost, QLD, Australia, 27–29 March 2012, pp. 245–249. New York: IEEE.
[21] Gander L, Lezuo C and Unterweger R. Rule based document understanding of historical books using a hybrid fuzzy classification system. In: *2011 Workshop on historical document imaging and processing*, Beijing, China, 16–17 September 2011, pp. 91–97. New York: ACM.
[22] Meier B, Stadelmann T, Stampfli J et al. Fully convolutional neural networks for newspaper article segmentation. In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, Kyoto, Japan, 9–15 November 2017, pp. 414–419. New York: IEEE.
[23] Tao X, Tang Z, Xu C et al. Logical labeling of fixed layout PDF documents using multiple contexts. In: *2014 11th IAPR international workshop on document analysis systems*, Tours, 7–10 April 2014, pp. 360–364. New York: IEEE.
[24] Antonacopoulos A, Clausner C, Papadopoulos C et al. Competition on recognition of documents with complex layouts-RDCL2015. In: *2015 13th IEEE international conference on document analysis and recognition*, Tunis, Tunisia, 23–26 August 2015, pp. 1151–1155. New York: IEEE.
[25] Shafait F, Keysers D and Breuel TM. Performance comparison of six algorithms for page segmentation. In: *International workshop on document analysis systems VII*, Nelson, 13–15 February 2006, pp. 368–379. Berlin: Springer.
[26] Suzuki S. Topological structural analysis of digitized binary images by border following. *Comput Vis Graph Image Pr* 1985; 30(10): 32–46.
[27] Otsu NA. Threshold selection method from gray-level histograms. *IEEE T Syst Man Cyb* 1979; 9(1): 62–66.
[28] Saad RSM, Elanwar RI, Kader NS et al. BCE-Arabic-v1 dataset: Towards interpreting Arabic document images for people with visual impairments. In: *9th ACM international conference on pervasive technologies related to assistive environments*, Corfu Island, 29 June 2016, pp.25–32. New York: ACM.
[29] Devlin J, Chang MW, Lee K et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *The North American chapter of the association for computational linguistics human language technology conference (NAACL-HLT)*, Minneapolis, MN, 2–7 June 2019.