International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-021-00382-4

ORIGINAL PAPER

Extracting text from scanned Arabic books: a large-scale benchmark dataset and a fine-tuned Faster-R-CNN model

Randa Elanwar[1] · **Wenda Qin**[2] · **Margrit Betke**[2] · **Derry Wijaya**[3]

Received: 3 July 2020 / Revised: 13 April 2021 / Accepted: 17 June 2021

Abstract

Datasets of documents in Arabic are urgently needed to promote computer vision and natural language processing research that addresses the specifics of the language. Unfortunately, publicly available Arabic datasets are limited in size and restricted to certain document domains. This paper presents the release of BE-Arabic-9K, a dataset of more than 9000 high-quality scanned images from over 700 Arabic books. Among these, 1500 images have been manually segmented into regions and labeled by their functionality. BE-Arabic-9K includes book pages with a wide variety of complex layouts and page contents, making it suitable for various document layout analysis and text recognition research tasks. The paper also presents a page layout segmentation and text extraction baseline model based on fine-tuned Faster R-CNN structure (FFRA). This baseline model yields cross-validation results with an average accuracy of 99.4% and F1 score of 99.1% for text versus non-text block classification on 1500 annotated images of BE-Arabic-9K. These results are remarkably better than those of the state-of-the-art Arabic book page segmentation system ECDP. FFRA also outperforms three other prior systems when tested on a competition benchmark dataset, making it an outstanding baseline model to challenge.

1 Introduction

Text extraction research has been pursued for decades, achieving almost perfect results for certain tasks, document domains, and languages when tested on limited-size datasets. With the availability of much larger datasets and deep learning models, the range of tasks and domains that are being addressed in document image research has widened significantly. Instead of separating text from non-text components in simple page layouts, the research challenges are now to extract text in book illustrations or tables, handling scanned versus born digital files, and interpreting documents in a low-resource language (Fig. 1).

Collecting a large dataset of scanned documents for document analysis research is a time-consuming task that typically requires large institutional funds and expert collaboration to collect, filter, and organize the documents, design an annotation scheme, and then manually provide annotations for each document in order to establish the ground truth needed to train supervised learning models (more details in the appendix sections 9.1, 9.2). Consequently, there is a lack of publicly available datasets of scanned documents in many low-resources languages. This was also the case for Arabic until Saad et al. provided BCE-Arabic-v1 [53], a dataset containing 1833 images of pages scanned from 180 books. We consider BCE-Arabic-v1 a precursor of BE-Arabic-9K, the dataset we introduce here. BE-Arabic-9K contains more than 9000 high-quality scanned images from over 700 Arabic books (Fig. 1). Among these, 1500 documents have been segmented into layout regions, and each region has been labeled by its content type (e.g., text or non-text).

We designed the collection process of BE-Arabic-9K, so that its images are useful as training data for various document analysis tasks. Foremost, we were interested in creating a dataset that supports research on the task of text extraction, that is, separating text from non-text compo-

✉ Randa Elanwar
randa.elanwar@eri.sci.eg ; eng_R_I_Elanwar@yahoo.com

Wenda Qin
wdqin@bu.edu

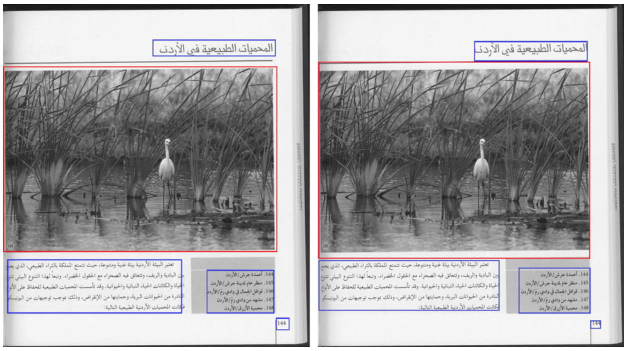
Margrit Betke
betke@bu.edu

Derry Wijaya
wijaya@bu.edu

[1] Computers and Systems Department, Electronics Research Institute, Cairo, Egypt

[2] Boston University, Boston, USA

[3] Department of Computer Science, Boston University, Boston, USA



المحميات الطبيعية في الأردن

المحميات الطبيعية في الأردن

Fig. 1 An image from BE-Arabic-9K segmented by FFRA system (left) and a human expert (right) with text regions bounded by blue boxes and non-text foreground regions by red boxes

nents of a document image (Fig. 1). Having a solution for this task, which is also called "text localization" or "page segmentation," is a basic requirement for most document analysis tasks. As a general purpose benchmark for Arabic documents, however, BE-Arabic-9k may also be used by other researchers to develop solutions for a wide variety of tasks, for example, physical layout analysis, logical layout analysis, table analysis, or diagram classification. "Physical layout analysis includes segmenting of the image into a set of non-overlapping homogeneous regions, called 'zones' or 'blocks,' and labeling each region according to its content class (text or graphic). Logical layout analysis (LLA) interprets the function of the text within the document (e.g., title, text body, caption, page number) and determines a reading order of the layout regions, which is required by PDF reading software for people with visual impairments" [28].

This paper offers a solution to the problem of a physical layout analysis of scanned Arabic book pages, which includes page segmentation into homogeneous regions and classification of these regions as text or non-text. The problem has been addressed in the literature, for example, for specific document domains like historical manuscripts [14–17] and newspapers [6,32], and a discussion of these approaches was provided in our previous work [28]. In 2018, we proposed the ECDP system [28], short for "Ensemble-based classification of document patches," as a general solution for analyzing scanned Arabic book pages. To the best of our knowledge, ECDP is still the state-of-the-art system for physical layout analysis of scanned Arabic book pages. ECDP uses a multi-step process that involves an ensemble of support vector machines and a voting mechanism to extract text.

To set up a baseline for the BE-Arabic-9K dataset, we developed a Faster-R-CNN-based model that leverages the advances that have been made in recent years with regard to deep learning solutions for region-based detection of objects in images. Faster R-CNN [51] is an efficient improvement of the "Fast R-CNN" model [30] and the earlier R-CNN method [31]. (R-CNN is short for "region with convolutional neural network features".) Experiments reveal that the fine-tuned Faster-R-CNN-based model outperforms ECDP [28] and three other systems [2,13,52] for extracting text from scanned Arabic book pages.
In summary, the contributions of our work are:

- Creating the 9000+ dataset BE-Arabic-9K by scanning pages from Arabic books with a wide variety of layout shapes and content and publicly sharing it.
- Annotating and releasing a subset of 1500 images for physical layout analysis research purposes.
- Presenting a deep learning baseline model FFRA based on fine-tuning pre-trained Faster-R-CNN structure for page segmentation.
- Showing high accuracy results for FFRA on the 1500 book page images, and also providing experimental comparisons of FFRA with four previous methods.
- Publicly sharing our FFRA model parameters and source code, as well as our ground-truth annotation interface code and our evaluation code at https://github.com/wdqin/BE-Arabic-9K

2 Background on datasets for Arabic document analysis

For research on Arabic documents analysis, past dataset collection efforts were mostly directed toward text-only documents, because the research questions focused on optical character recognition and handwriting recognition in Arabic. An example of a dataset of machine printed Arabic text is the Arabic Printed Text Image (APTI) dataset [56]. A large-scale dataset of handwritten documents is the MADCAT dataset (Multilingual Automatic Document Classification Analysis and Translation) [57], which contains 38,000 handwritten Arabic pages of news. The IFN/ENIT-database of Tunisian town names [47] contains 2200 handwritten forms from 411 writers and about 26,000 binary-word images. Another example is the ALTEC dataset [7], which consists of 5000 pages with approximately 35,000 lines (around 175,000 words and 1 million characters). The KHATT database [41] contains 2000 random text paragraphs consisting of 9327 lines written by 1000 distinct writers. Handwriting datasets have also been created by recording user interactions with electronic devices, i.e., recording the trace of a stylus pen, while the user is writing on an electronic surface like a tablet or smartphone. Examples are the online version of the KHATT dataset [42], the ADAB dataset [25], and the ALTEConDB dataset [1].

In contrast to the richness of text-only research datasets, large-scale publicly available datasets for page segmentation and layout analysis of scanned documents in the Arabic

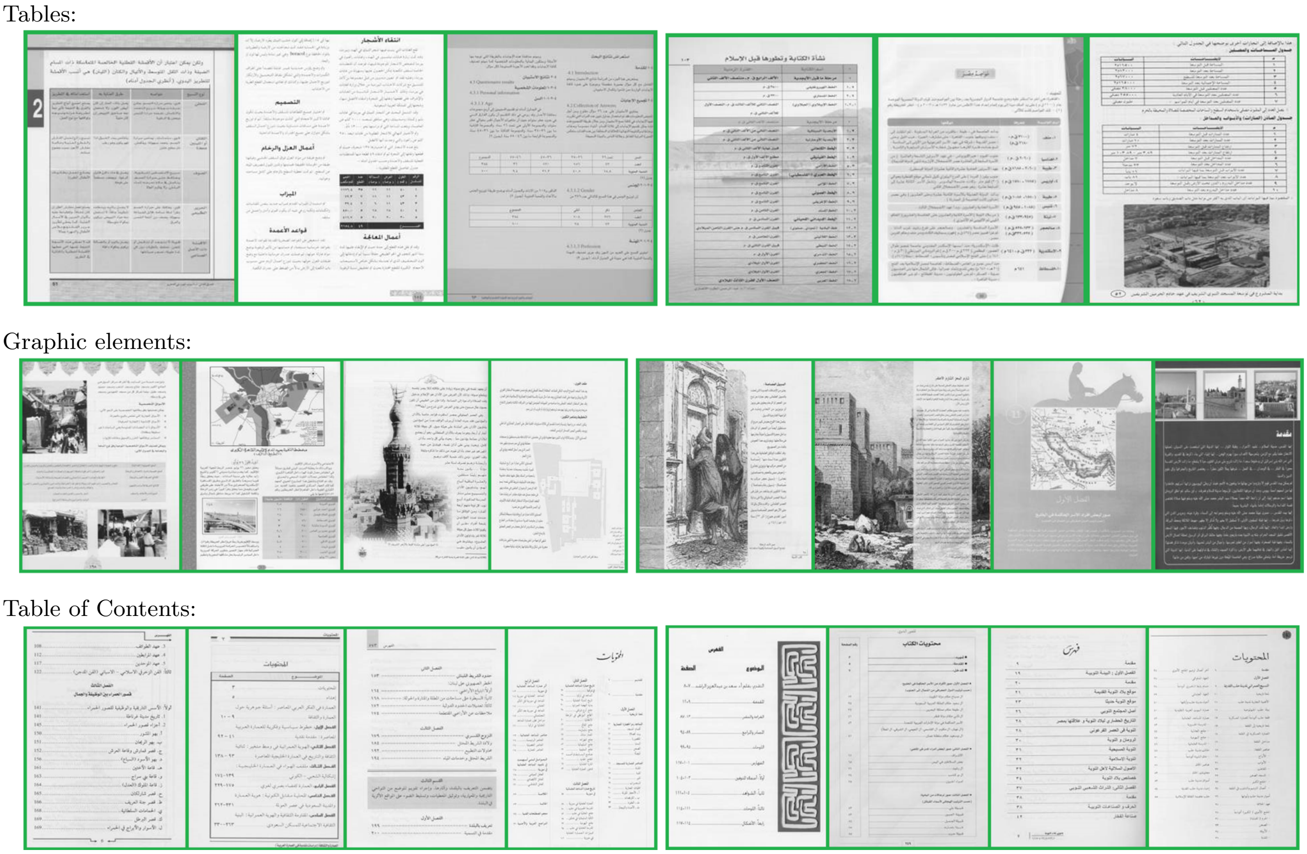


Fig. 2 Tables could contain text only and sometimes contain graphic elements. The text in tables could be in the same language or multilingual. The text direction could vary horizontally and vertically. The layout could be one table covering the entire page or multiple tables cascaded horizontally or vertically. The text background varies as well from plain white background to variant colored or watermarked backgrounds and decorative frames and borders around specific logical text like titles and page numbers

language did not exist until Saad et al. [53] published BCE-Arabic V1 (1833 images). There are only two other datasets (available upon request), both small scale, that include document images from newspapers with complex layouts (50 images) [6] and historical books (85 images) [35]

3 Background on Arabic documents analysis solutions

Separating text components from non-text in a document image is one basic requirement for almost all document analysis tasks. Many classical ad hoc algorithms were proposed until the early 2000s followed by supervised machine learning solutions. Such solutions require training samples with annotations including each region's type and its bounding box information. Accordingly, Arabic as one example of low-resource languages is not well represented in the literature, compared to other languages, in terms of benchmark datasets or baseline results. Most works for Arabic were dedicated to historical documents [11,36] (a comprehensive survey of challenges of the historical Arabic documents processing and the existing solutions can be found in [37]) and very few more to newspaper images. The related works all depend on well-known models like neural networks (NN), support vector machines (SVM), decision trees, or random forests (RF). The target of the developed solutions was either to detect homogeneous text regions or segment textlines directly.

The recent deep-learning-based solutions developed for text detection were created using structures like fully convolutional neural networks (FCNN) and its variants, convolutional encoder-decoder (CED), etc. Earlier attempts to use deep structures for Latin scripts documents deployed them as unsupervised feature extraction stages and left the classification task of feature maps obtained to a separate softmax or support vector machine models as in [19,20,46,60]. At this stage, only model training from scratch was performed to few convolutional layers.

Later attempts started to use the deep models as an end-to-end system with little or no preprocessing to the input image and getting the final output directly from the model's top layer. The models got bigger in size (i.e., deeper), and data augmentation took place to support the computation of the increased hyperparameters.

The recent study by Studer et al. [58] concluded that whether it is training-from-scratch or cross-domain learning from a pre-trained model, the results of semantic segmentation problem depend on the test dataset. Some datasets could be segmented with high accuracy, and some are not regardless of the model structure or the training method. To come with these conclusions, the authors initialized their models encoder with the pre-trained weights from ImageNet and compared their performance, with the same models trained from scratch using historical document dataset (DIVA-HisDB).

Recently, Faster R-CNN is frequently used with document analysis research datasets, especially for logical text detection. In [68], a fine-tuned Faster R-CNN acted as the baseline model for detecting logical regions such as titles and tables in the PubLayNet documents dataset images. Similarly, for Doc-Bank dataset a Faster-R-CNN-based model was deployed for detecting more subtle regions like abstract, caption, etc. [43] For Arabic language very few attempts for text detection in document images using deep structures were reported. Amer et al. used CNN to classify regions previously segmented by ARLSA algorithm [45] as ‘text’ or ‘non-text’ regions in newspapers images, while Barakat et al. [12,13] used FCN models for pixel classification to detect handwritten text lines in Arabic historical documents. Recently, Neche et al. [44] used RU-Net to classify pixels as ‘textline’ or ‘background’ for segmenting Arabic handwritten documents and then used a pipeline of CNN-BLSTM-CLC to further segment the detected textlines into words. In 2014, [31] proposed a region-based object detection method called region with CNN features (R-CNN). Since then, various improvements were built based on R-CNN. [30] proposed a more efficient solution called Fast R-CNN. In [51], a deep learning network called Faster R-CNN was proposed to make block-wise object detection more efficient. In the work of Mask R-CNN [33], the network was further extended to detect objects in pixel level. Most recently, Almutairi et al. [5] proposed a mask R-CNN-based model attempting to solve the segmentation of newspaper contents at a semantic level. However, the solution does not provide a sufficient quantitative analysis of any public dataset, which is one of our attempts in this paper. Also, the evaluation of the system is based on average precision and model loss, which is not explanatory enough to show how well the segmentation that the system can perform. To the best of our knowledge, we are the first to provide text/non-text block segmentation using Faster R-CNN for an Arabic documents dataset and to provide the model evaluation on block level inspired by [54], as described in Sect. 6.3. Accordingly, we choose to adopt Faster R-CNN, which is a block-based method, as our basic model for segmenting book pages in Arabic.

4 BE-Arabic-9K: dataset collection, contents, and annotation process

To make the BE-Arabic-9K dataset suitable as a training data and evaluation benchmark for multiple Arabic document analysis tasks, we foremost concentrated on the scale of the dataset and provided more than 9000 digitized Arabic book pages. We conducted the scanning process at three university libraries with Arabic book collections, Boston University, Harvard University, and Massachusetts Institute of Technology, over a period of several weeks. We considered a wide range of book topics, such as literature, science, mathematics, arts, politics, and religion, and ensured that the selected book pages include a wide range of non-text elements like illustrations, charts, decorations, tables, equations, line drawings (pen sketches), and music notes, in addition to various non-uniform background textures. By digitizing the 9000 book pages from more than 700 books by different publishers, located in the Middle East, North Africa, and the Gulf Region, and with various publishing dates, covering a long period, we ensured that BE-Arabic-9k’s content is generic and representative to the most common printing fonts and styles in different regions over long time period.

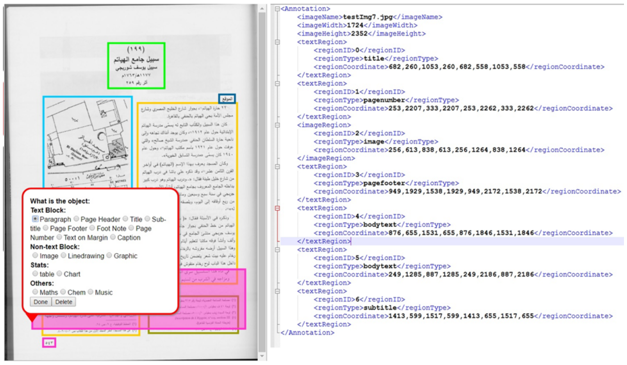
BE-Arabic-9k contains samples from books published between 1956 and 2016 written by 419 authors from 244 unique publishers in 22 countries. All the details can be found in the dataset index sheet here([1]). Note that some books had missing information we could not add to these statistics.

To ensure a high-quality digitization effort, we scanned the book pages using scanners particularly suited for books, rather than loose-leaf document scanners, which researchers typically use for data collection. We note that the use of loose-leaf document scanners to digitize book pages often leads to imperfections in the resulting document image like severely skewed or warped text, noisy page borders, salt-and-pepper noise, etc. The specialized books scanners’ embedded software does good job in general for skew and warp correction beside splitting adjacent images but this is not guaranteed most of the time, especially with the relatively old prints. The data still have challenging imperfections nevertheless incomparable to what might be introduced by regular document scanners. We scanned the book pages at 300 or 600 dpi in grayscale rather than black and white, regardless of the image file sizes, to maintain the image quality and avoid distortion.

To support the annotation process of BE-Arabic-9K, we created a dedicated interface (Fig. 3). We modified the LabelMe tool [39], which enables a user to provide polygonal segmentation of objects in images, to our purpose. Our interface displays the Arabic book page image and enables the annotator to segment different regions by drawing rectangles

---

[1] https://github.com/wdqin/BE-Arabic-9K.


What is the object:
Text Block:
● Paragraph ○ Page Header ○ Title ○ Sub-title ○ Page Footer ○ Foot Note ○ Page Number ○ Text on Margin ○ Caption
Non-text Block:
○ Image ○ Linedrawing ○ Graphic
Stats:
○ table ○ Chart
Others:
○ Maths ○ Chem ○ Music
Done
Delete

```
<Annotation>
    <imageName>testImg7.jpg</imageName>
    <imageWidth>1724</imageWidth>
    <imageHeight>2352</imageHeight>
    <textRegion>
        <regionID>0</regionID>
        <regionType>title</regionType>
        <regionCoordinate>682,260,1053,260,682,558,1053,558</regionCoordinate>
    </textRegion>
    <textRegion>
        <regionID>1</regionID>
        <regionType>pagenumber</regionType>
        <regionCoordinate>253,2207,333,2207,253,2262,333,2262</regionCoordinate>
    </textRegion>
    <imageRegion>
        <regionID>2</regionID>
        <regionType>image</regionType>
        <regionCoordinate>256,613,838,613,256,1264,838,1264</regionCoordinate>
    </imageRegion>
    <textRegion>
        <regionID>3</regionID>
        <regionType>pagefooter</regionType>
        <regionCoordinate>949,1929,1538,1929,949,2172,1538,2172</regionCoordinate>
    </textRegion>
    <textRegion>
        <regionID>4</regionID>
        <regionType>bodytext</regionType>
        <regionCoordinate>876,655,1531,655,876,1846,1531,1846</regionCoordinate>
    </textRegion>
    <textRegion>
        <regionID>5</regionID>
        <regionType>bodytext</regionType>
        <regionCoordinate>249,1285,887,1285,249,2186,887,2186</regionCoordinate>
    </textRegion>
    <textRegion>
        <regionID>6</regionID>
        <regionType>subtitle</regionType>
        <regionCoordinate>1413,599,1517,599,1413,655,1517,655</regionCoordinate>
    </textRegion>
</Annotation>
```

Fig. 3 Annotation interface for region segmentation and labeling (left). Output XML file (right)

around them. This can be done quickly by using two mouse clicks. A pop-up menu then appears with a list of labels to choose from. For the purpose of automated text extraction, two labels would have sufficed (text and non-text). However, since BE-Arabic-9K is designed to support multiple Arabic document analysis tasks, we opted to provide a list of labels that enables logical layout analysis, i.e., determines the function of the text within the document (e.g., title, text body, caption, page number). The interface with an annotated book page image and the output XML file that stores the annotation results for this image are shown in Fig. 3.

In our pursuit to provide ground-truth labels for the BE-Arabic-9K data, we were inspired by the success of crowdsourcing in supplying reliable annotations of research datasets at low cost and in a short time. For example, crowdsourcing was previously used to segment and label image datasets like the popular ImageNET dataset [24] and text datasets in Arabic [22,63,66,67]. (More details can be found in the appendix section 9.3.) The challenge was to design micro-tasks that tempted competent and reliable crowd workers to accept these tasks and deliver accurate results at acceptable costs. To prepare for the crowdsourcing experiment, we integrated our annotation interface with the popular crowdsourcing platform Amazon Mechanical Turk (MTurk) [8]. In a pilot experiment with 725 book pages that were annotated by three different crowd workers per page, we found a high variance in the way workers segmented the regions of interest. Some workers did not follow the instructions to create close-fitting bounding boxes and included large areas of background in the segmentation. Others overlooked small text regions, for example that included page numbers, did not take care of small spaces between different page components, or drew overlapping rectangles. We also encountered a "spam worker" who submitted random and incomplete jobs very quickly. Based on this outcome of the pilot experiment, to train and test our model, we decided to provide the ground-truth annotations of the text and non-text regions of the documents ourselves.

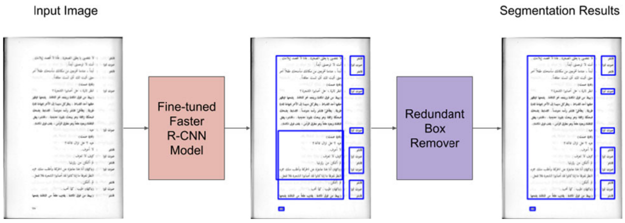


Fig. 4 Workflow of the FFRA model

5 Fine-tuned Faster R-CNN on Arabic" (FFRA)

In this section, we present our fine-tuning of the pre-trained Faster R-CNN model structure as the baseline model for text regions extraction in BE-Arabic. For the convenience of reading, we call this baseline model FFRA, which stands for **F**ine-tuned **F**aster **R**-CNN on **A**rabic. FFRA takes as input a color or grayscale image of a scanned Arabic book page and yields as output the bounding boxes of the text and non-text regions of the page. FFRA is a two-step system. The first step passes the original image through a fine-tuned Faster R-CNN network [51], which produces numerous, potentially overlapping candidate text and non-text regions. The second step is to remove the redundant regions. The remaining regions are the segmentation results of FFRA. A visualization of the FFRA workflow is shown in Fig. 4.

5.1 FFRA component: fine-tuned Faster R-CNN

The Faster R-CNN model [51] can detect and classify, by default, 91 types of objects in an image, e.g., airplanes or dogs, as well as the image background.

We changed the number of object classes from 91 to the three classes relevant here, text, non-text, and image background to suit our classification problem.

The Faster R-CNN model consists of three components. The first component is a group of convolutional layers that are used to extract feature maps from the input image. For our fine-tuned model, we adopted the pre-trained layers of the ResNet-50 model [34] as our initial shared convolutional layers.

The outputs of these layers are used by the other two components of the Faster R-CNN model. The second component of R-CNN is a sub-network called Region Proposal Network (RPN). It proposes a set of potential rectangular regions that are likely to contain "objects," of interest, in our case text or non-text foreground areas. The proposed regions are represented by their bounding box coordinates (i.e., "box regression" output) and the class number of the predicted object inside. Although it includes a verification and refining mechanism, the RPN is not able to interpret foreground "objects" as text or non-text regions, a task that is performed here by a Fast R-CNN model [30]. The Fast R-CNN com-

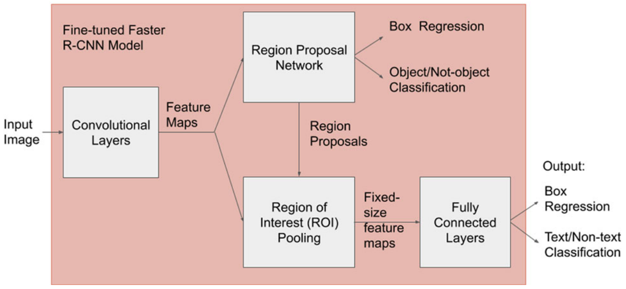


Fig. 5 Network architecture of the fine-tuned Faster R-CNN. Outputs from the fully connected layers are the predicted positions of text and non-text regions (*box regression*), and the predicted labels of boxes (*box classification*)

ponent starts with a pooling layer called "Region of Interest (ROI) Pooling," which extracts the proposed regions in the feature maps and converts these regions of different sizes into fixed-size regions. The fixed-size regions are then passed to a group of fully connected layers. Eventually, these fully connected layers produce two kinds of outputs: the class type and location of each object in the format of a bounding box, here the locations of the text and non-text regions. Each output box is accompanied with a "confidence score" that reports the model's confidence in the detection. The architecture of the Faster R-CNN model used by FFRA is given in Fig. 5.

5.2 FFRA component: redundant region removal

A pilot experiment on a small subset of our data showed that Faster R-CNN model was not promoting the selection of few non-overlapping region proposals. For example, a case of "over-segmentation," i.e., creating redundant overlapping text regions, can be seen in the middle of Fig. 4, where the main text on the page is incorrectly separated into two regions. We found many other cases where the Faster R-CNN model created multiple overlapping bounding boxes that belong to the same text or non-text region of the document image. A reason why so many overlapping regions are produced by the Faster R-CNN model may be that unlike a concrete object in a scene image like a dog, the sub-regions of a text area are still text areas (while subimages of a dog cannot fully represent the dog). This is also true for some of the non-text areas—a part of a cartographic map, for example, could still be recognized as a unique cartographic map. Moreover, our image document analysis task is challenging because the sizes of the regions to be detected vastly differ from each other.

The overlapping bounding boxes need to be reduced to a single box since the system should not extract the same text twice (e.g., as part of reading software for people with visual impairments). Redundant regions are considered to be over-segmentation errors when matched with a ground-truth page layout. If we simply merge these overlapping bounding boxes, the chance of producing under-segmentation errors increases.

We propose a solution here that removes redundant bounding boxes by avoiding obvious over- and under-segmentation errors. The method is rule based: Only boxes that have a confidence score of at least $\tau_c = 0.8$, computed by the fine-tuned Faster R-CNN, are considered. For each pair of bounding boxes that overlap each other and belong to the same class, the ratio of the intersection area to the area of the smaller bounding box is computed. If the ratio is higher than a threshold τ_o, the two bounding boxes are merged into a single bounding box so that its coordinates are given by the left-most, upper-most, right-most and bottom-most value of the vertices among the two bounding boxes. Our validation experiment showed that a heuristic value of $\tau_o = 0.2$ provides a balance between avoiding an over-segmentation error caused by not merging redundant bounding boxes, and avoiding an under-segmentation error, caused by merging bounding boxes that belong to different same-class segments. A visualization of the result after the merging process is provided in the right document image of Fig. 4. By comparing the document at the center of the figure and the document on the right, we can notice that appending the redundant box remover component merges the overlapping bounding boxes obtained by the Faster R-CNN to reduce prediction redundancy.

6 Experiments and results

We performed two main sets of experiments to evaluate the performance of FFRA and compare it to prior work. The first set of experiments uses images in BE-Arabic-9K. The second set of experiments tested FFRA on three sets of images in the ASAR challenge [27].

A final experiment is performed to compare the Faster R-CNN structure performance to one of the latest versions of another object detection deep structures YOLO4 [4]

6.1 Experimental data

We selected 1500 images of BE-Arabic-9K and 300 images of BCE-Arabic-v1 [53] as our main experimental dataset. (We included the latter because prior work was trained on BCE-Arabic-v1.) This 1800-image collection is a representative sample of Arabic book pages from various countries and periods with the following statistics:

- 1598 pages with text in various fonts and font sizes within a variety of layout elements such as paragraphs, page numbers, titles and subtitles, headers, and footnotes, and no illustrations;

- 255 pages in double-column format with a special text-line reading order (Arabic poetry, play narration, etc.);
- 36 pages with text with decoration elements (frames, ornaments, etc.);
- 6 pages with text with tables;
- 12 pages with text formatting of a book index, such as table of contents, glossary, or references;
- 2 book covers;
- 19 pages with book section separators, such as chapter title and illustration, or a list of chapter and subsections titles;
- 202 pages with text and non-text illustrations such as photographs, line drawings, maps, and charts and diagrams.

The 1800 images contain 7088 regions annotated as text and 278 regions annotated as non-text regions. On average, each image contains 4.1 foreground regions (3.94 text and 0.15 non-text regions).

To motivate the document image research community to develop automatic solutions for providing annotation for scanned Arabic book pages, we organized a Physical Layout Analysis Challenge [9] at the IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR 2018) [10]. We used the 90-image ASAR challenge dataset [27] as a secondary experimental dataset to test FFRA. The dataset contains three sets of 30 images with different layout challenges, i.e., text and picture regions in a single column layout (set A), a double column layout (set B), and a format that is neither single-column nor double-column (set C). On average, each image contains 12.14 foreground regions.

6.2 FFRA training methodology

The 1800-image dataset described above was divided into 6 groups, with 300 book pages in each group. One group was used as a validation set for tuning the hyper-parameters τ_c and τ_o of the FFRA model. During the validation experiment, the five groups of images that were not the validation set were used as the training data, and the validation set as the testing data. After tuning the two hyper-parameters, the five training groups in the validation experiment were used in a 4-to-1 round-robin manner for fine-tuning of the weights of the Faster R-CNN model and then testing to evaluate the entire FFRA system.

Because the size of the training dataset is relatively small to fine-tune a deep learning network, we applied data augmentation to our training data. We used a public library called “Albumentations” [18] and the “transforms” part of the “torchvision” detection package in PyTorch [49] to implement the data augmentation. In particular, we performed Gaussian blurring with a maximum kernel size of 3, randomly changing the brightness up to a factor of 0.1, and randomly changing the contrast up to a factor of 0.5.

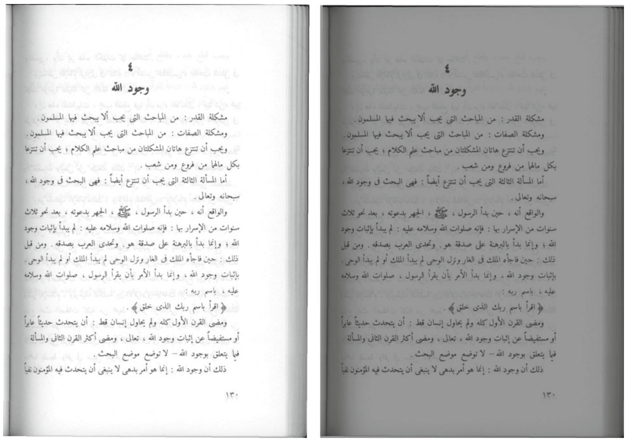


Fig. 6 Augmentation of training data: By changing the contrast and brightness of the original image (left), we created training data (right) that help model “scanning effects” due to lighting variations

We repeated this data augmentation ten times to produce ten different augmented images from the original one. Additionally, using the torchvision code, we horizontally flipped the image and the ground truth bounding boxes on the 10 augmented images. An example of an augmented image is shown in Fig. 6.

The ground-truth annotations of the 1800 book page images were provided in PAGE XML format [48]. We converted the coordinates and the labels of each text and non-text bounding box into numeric vectors and used them as the targets for the loss computation of the FFRA model. We used the pre-trained Faster R-CNN model with a ResNet-50 backbone as provided in Pytorch as the first component of our FFRA model. We used Stochastic Gradient Descent (SGD) for optimization with a learning rate of 5×10^{-3}, momentum of 0.9, and a weight decay of 5×10^{-4}. During training, we reduced the learning rate of each parameter group by 0.1 every 3 epochs. We trained 10 epochs for a single model. Our training GPU was a RTX 2080 Ti, each training epoch is finished in approximately 25 min. We used OpenCV 3 for image processing and removing of the redundant bounding boxes in the second component of FFRA. We share our model parameters and source code at[2].

6.3 FFRA evaluation methodology and benchmark comparisons

The segmentation results of FFRA and prior work were evaluated with regards to (1) image regions and (2) pixels. The region-based evaluation process was also used in the ASAR physical layout analysis challenge [27]. It was inspired by

---

[2] https://github.com/wdqin/BE-Arabic-9K.

previous work [54] and uses the term “block” to define metrics with regard to the image foreground regions:

- The correct-segmentation (CS) rate of an image is the number of correctly segmented blocks that match the corresponding blocks in the ground truth, normalized by the number of ground truth blocks in the image.
- An over-segmentation occurs when a foreground block is not detected as a single block but split into two or more blocks. The over-segmentation error (OSE) rate of an image is the number of additional blocks, normalized by the number of ground truth blocks in the image.
- An under-segmentation occurs when two foreground blocks in the ground truth are detected as a single foreground block. The under-segmentation error (USE) rate of an image is the number of under-segmented blocks divided by the number of ground truth blocks.
- The missed-segmentation error (MSE) rate of an image is the number of missed blocks, normalized by the number of ground truth blocks (“false negative detection rate”).
- The false alarm error (FA) rate is the number of segmented blocks that are not found in the ground truth, normalized by the number of ground truth blocks (“false positive detection rate”).
- The overall block error rate ρ is the summation of OSE, MSE, and USE rates.

To report segmentation results at the pixel level, we compute two metrics, also used in a previous work [61] that compare the predicted and ground truth labels of each pixel: Foreground Pixel Accuracy: FgPA = $\sum_{x \in I_f} \mathbb{1}(x)/n_f$, Total Pixel Accuracy: TPA = $\sum_{x \in I} \mathbb{1}(x)/n$,
where n is the number of pixels in the image I and n_f is the number of pixels in the foreground regions I_f of image I (i.e., I_f contains all the pixels in text and non-text regions). The indicator function $\mathbb{1}(x)$ is 1 if the labels of pixel x match and zero otherwise.

FFRA was compared with four prior works, ECDP [28] RFAAD [52], an FCN-based method [13], and an adaptive thresholding method [2]. ECDP is the state-of-the-art system for physical layout analysis of scanned Arabic book pages. It was trained and tested using BCE-Arabic V1 [53].

ECDP [28] uses a multi-step process that involves an ensemble of five support vector machines and a voting mechanism to classify image patches, represented by edge and Fourier transform features, into text and non-text classes. Resulting patches of the same class that are adjacent to each other or overlapping are combined into larger image regions, yielding rectangular foreground boxes that are labeled as text or non-text. For comparison, ECDP was tested using the same fivefold sets in round-robin cross-validation scheme as FFRA. Results were reported as averages of the five experiments (Table 1).

RFAAD [52] detects small connected components on the Arabic book page image, performs morphological operations to merge them, and then extracts geometric features from the merged connected components to train a random forest model.

The FCN-based method [13] trains a fully convolutional network [40] to perform object segmentation, with pre-trained layers of VGG-16 network [55].

The adaptive thresholding-based method [2] uses a flexible threshold value for the binarization of the input image and followed by morphological operations to obtain regions of interests, which are then classified by heuristic rules to determine whether they are text or non-text regions.

To make sure our experiments investigate the top performing deep structures for object detection, we also considered an additional comparison of our baseline model to another method, namely YOLO [4]. The earlier releases of YOLO (v2 and v3) were known to have struggled with small objects within the image due to the spatial constraints of the algorithm.

Since we have small size text areas in our dataset images which represent important logical information like page numbers, section numbering and text on margins, we assumed the algorithm will be of limited performance. Few months ago YOLO4 was released so we evaluated it against our dataset to investigate its performance.

The experiments for YOLO were performed by replacing the middle component of FFRA with pre-trained YOLO4 structure instead of Faster RCNN and fine-tuning on our BE-Arabic-9k dataset.

Fine-tuning a YOLO structure requires a lot of training epochs to obtain acceptable results. Therefore, both structures were trained for 100 epochs to update the weights, keeping the default hyper-parameters of each structure unchanged: Backbone (ResNet-50 for Faster R-CNN and yolov4.conv.137 for YOLO), optimizer (SGD for Faster R-CNN and Adaptive Moment Estimation (Adam) for YOLOv4). We cross validated both structures using our 1500 images and the results are shown in Table. 3 We can see that the Faster R-CNN is around 1–2% better than YOLOv4 regarding pixel-wise classification results while relatively the same for the rest. We discover that it is more likely for YOLOv4 to detect only parts of a large text/non-text block while not covering the whole region. This might be a potential reason why fine-tuned YOLOv4 has a lower pixel-wise classification performance than the Faster R-CNN ones.

As such ”partial segmentation” error applies to both Faster R-CNN and YOLOv4 network, which could affect the performance of the whole system, we think this is an interesting and promising topic for future investigation.

Table 1 Experimental results on a subset of BE-Arabic-9K and BCE-Arabic-v1

		Block Segmentation						Block Classification		Pixel Classification	
		CS	OSE	USE	MSE	FA	ρ	F1	Accuracy	FgPA	TPA
ECDP [28]	Avg.	1.06	0.2	2.14	0.16	0.38	0.56	91.06%	92.51%	93.78%	94.94%
FFRA	Avg.	**2.29**	**0.08**	**0.93**	**0.07**	**0.03**	**0.22**	**99.15%**	**99.38%**	**96.60%**	**97.23%**

Error rates are reported as averages over 1500 images. The results on block segmentation include text and non-text foreground regions. The classification results denote percent accuracy in text versus non-text prediction of foreground blocks or pixels. Standard deviations for the reported segmentation rates are less than 0.1 and for classification scores less than 1 percent point

Bold highlights the maximum results achieved per each aspect of comparison

Table 2 Experimental results on the ASAR challenge datasets [27], reported as averages over 30 images

	Foreground block segmentation						Block classification		Pixel classification
	CS	OSE	USE	MSE	FA	ρ	F1	Accuracy	TPA
					ASAR Set A				
RFAAD [52]	**10.10**	1.37	**1.00**	**0.43**	2.50	**2.80**	82%	75%	69%
FCN-Based [13]	9.06	3.67	1.94	2.43	1.83	8.04	88%	**97%**	80%
Adap. Thr. [2]	6.07	8.50	3.20	1.40	6.10	13.10	**93%**	90%	89%
FFRA	7.43	**1.13**	2	3.2	**0.2**	6.33	89%	85%	**97%**
					ASAR Set B				
RFAAD [52]	**13.25**	1.36	**1.68**	**1.07**	3.50	**4.10**	79%	71%	59%
FCN-Based [13]	9.13	3.07	4.13	2.83	1.04	10.03	**97%**	**99%**	87%
Adap. Thr. [2]	6.5	15.47	4.57	1.23	6.5	21.27	82%	87%	86%
FFRA	10.33	**0.57**	5.03	2.97	**0.27**	8.57	89%	86%	**99%**
					ASAR Set C				
RFAAD [52]	**9.40**	0.66	**0.59**	1.45	3.60	**2.70**	77%	69%	71%
FCN-Based [13]	6.90	3.79	0.86	1.83	1.73	6.48	**90%**	**93%**	75%
Adap. Thr. [2]	6.53	5.93	3.07	**0.93**	2.97	9.93	76%	82%	82%
FFRA	8.33	**0.2**	4.27	2.23	**0.07**	6.7	**90%**	87%	**92%**

Bold highlights the maximum results achieved per each aspect of comparison

Table 3 Validation results on different fine-tuned object detection models

	Block segmentation						Block classification		Pixel classification	
	CS	OSE	USE	MSE	FA	ρ	F1	Accuracy	FgPA	TPA
Faster R-CNN	**2.27**	**0.04**	**1.07**	**0.09**	0.04	**0.24**	98.80 %	99.08%	**97.22%**	**96.59%**
YOLOv4	2.09	0.07	1.12	0.15	**0.03**	0.28	**99.53%**	**99.32%**	94.51%	95.52%

6.4 Results and discussion

The results of FFRA and the state-of-the-art ECDP in the fivefold cross-validation experiments described above are reported in Table 1, and the results of FFRA and the three ASAR challenge methods on the ASAR datasets are reported in Table 2.

Our experimental results show that our FFRA has better performance than ECDP in almost every aspects of segmentation and classification according to seven metrics. For the three metrics for which ECDP outperforms FFRA, the difference in rates is almost negligible (no more than 0.1).

FFRA also shows strong performance on the ASAR data. It is the best model at pixel classification and second-best at block classification. We note that both the two top block classification models, "FCN-based method" and FFRA, each have a deep structure, which may be the reason that they outperformed the rule-based adaptive thresholding method [2] and the traditional learning method [52]. For block segmentation, we found FFRA has the lowest over-segmentation error rate while having relatively large under-segmentation and missed-segmentation error rates. This likely due to the fact that FFRA was trained and tuned on data with 4.1 foreground regions per image, and tested on data with 12.14 regions per image (in the ASAR annotations, large text regions are split

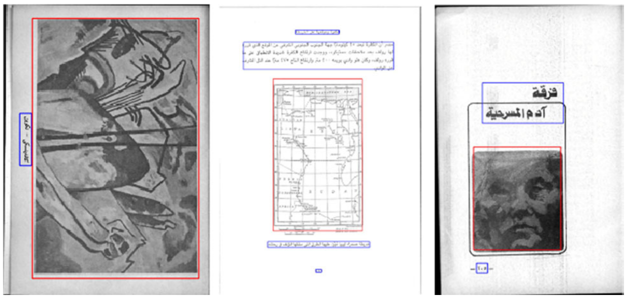


Fig. 7 Examples of successful segmentation by FFRA

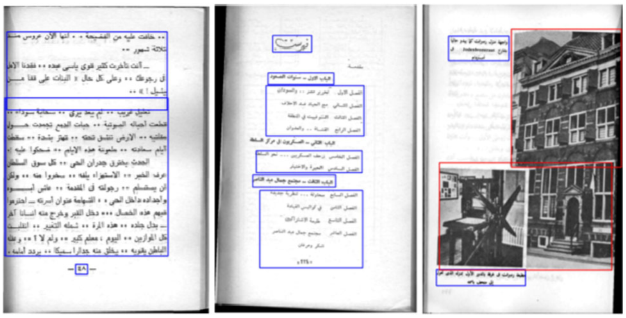


Fig. 8 Examples of erroneous segmentation by FFRA: A single paragraph was oversegmented (left), two text regions were missed (middle), and overlapping images were unavoidably outlined incorrectly due to the constraint that rectangular foreground blocks cannot overlap (right)

into small multiple paragraphs). With regard to the number of correct segmentations and combined error rate ρ on the ASAR data sets, FFRA is competitive and shows generalization ability.

We provide sample outputs of FFRA that visualize its ability to handle several challenges that are difficult for previous methods (Figs. 1, 4, 7, and 8). In particular, FFRA can segment images with brightness variations, which are common in scanned documents due to scanner quality differences. This includes the first image in Fig. 7, which has a large dark foreground region, a dark background region, and is blurry. Another challenge FFRA overcomes is recognizing non-text regions that do not have clear boundaries like the map in Fig. 7, middle, and text regions that have extraneous markings that could confuse an automated method like the two-line text region that has a distracting edge of a black, round-corner rectangle overlaid in Fig. 7, right. Success in these cases may be partially attributed to the fact that FFRA processes the full intensity information of the document image while traditional methods lose information in their binarization process. Furthermore, as a deep model, FFRA can learn a representation of the notion of “Arabic text” and distinguish it from line drawings such as the map in Fig. 7, middle.

We also show some cases where FFRA makes mistakes, for example, oversegments a paragraph, misses some text regions, or fails to find the outline of pictures correctly (Fig. 8). A human annotator would not have difficulties in detecting the text and image outlines in the images in Fig. 8. Knowledge of Arabic, however, would be needed to determine that the text in Fig. 8, left, belongs to a single paragraph.

7 Conclusions and future work

BE-Arabic-9K may become a solid foundation for building Arabic language resources for computing researchers. It is the first of its kind and may be used to serve a digitally underresourced but large community of 300 million people who speak Arabic or use Arabic characters to represent their languages, e.g., Amharic, Hausa, Kurdish, Farsi, Pashto, Swahili, Urdu, or Wolof. BE-Arabic-9K is the first large-scale dataset of scanned Arabic book pages suitable for training and testing deep learning models designed to solve various document analysis problems. The dataset is unique not only due to its size but also due to the wide variety of content and layout of pages collected from books published in the Middle East, North Africa, and the Gulf Region at various times during the last few decades. The dataset will be enlarged and re-annotated over time to suit different needs of researchers. We hope that the current and future versions of BE-Arabic-9K, provided at[3] , will serve researchers to develop document-analysis solutions that assist people with visual impairments in the Arabic world.

Our deep model FFRA shows the benefit of applying deep learning to the task of Arabic document analysis. The experimental results convincingly show that Arabic text regions can be detected with very high accuracy. The segmentation problem, i.e., obtaining accurate outlines of non-text and text regions, would benefit from additional research. Systems, such as FFRA, may be used within a comprehensive document analysis tool that detects the functionality of the various parts of a document layout, determines a reading order, and converts text and non-text information into readable material for visually impaired people. To facilitate such future research, we make our FFRA model parameters and source code publicly available at [4].

As a future work, we would want to conduct more experiments using the other object detection methods YOLO after fine-tuning and compare it to FFRA performance against larger portion of our dataset. Our dataset is very challenging and has samples with both unique objects and complex layouts and we believe that applying one method would not be able to pass all the challenges. Combining multiple solutions is expected to enhance the detection results.

---

[3] https://github.com/wdqin/BE-Arabic-9K.

[4] https://github.com/wdqin/BE-Arabic-9K

Acknowledgements The authors would like to thank the library staff at the Mugar Library at Boston University, the Rotch Library at MIT, and the Widener and Fine Arts libraries at Harvard University for facilitating the collection process of our dataset BE-Arabic-9K. The authors thank the National Science Foundation (1838193) and the Hariri Institute for Computing at Boston University for partial support of this work.

Appendix: Documents images datasets

Collection and annotation

Large size annotated datasets are one crucial resource needed for supervised machine learning. Researchers often spend a considerable amount of time annotating their self-collected datasets because they cannot find a publicly available dataset that match their research need. This may result in them ending up conducting their research on limited size datasets.

Unless special characteristics are required for a research dataset, the data collection phase is not as challenging or as expensive as data annotation. For example, The internet archive has billions of **unlabeled** images that could be downloaded using web search crawlers, an approach that has been followed before to construct large public computer vision datasets like TinyImage [59] and ImageNet [23]. The annotation phase is what controls the research outcome. Annotations should match the research question, meaning that a single image can have multiple annotations and several levels of details. Accordingly, the annotation process has been always expert-based and problem-oriented, expensive and time consuming.

Dataset collection for document analysis and recognition is one of the most challenging tasks compared to other research areas. One might expect ”transcripts” as the only annotation needed for documents images, however, according to the research problem the required annotations might include much more information like:

1. Segmentation information: locating the text position inside the document image (i.e., bounding box coordinates),
2. Logical labeling: identification of the text logical function (i.e., title, caption, footnote, etc.)
3. the text reading order (specially in multi-columns layouts)
4. Geometrical labeling: classifying the non-text element type (i.e., image, chart, map, logo, math formulae, etc.)
5. Descriptions: the alternative text for image elements, cells functions-and-relations for tables elements.

Logical labeling of a document’s text elements, is one of the most human-intelligence-based tasks that sometimes become controversial especially with unfamiliar layouts or with the absence of appropriate text formatting (e.g., font size and emphasis). Our previous attempt to provide the first labeled dataset for page segmentation and layout analysis BCE-Arabic V1 was one of a kind [53]. We investigated the importance of having physically analyzed documents (i.e., segmenting regions and identifying their type as text or non-text), and showed that it is no trivial task and has a significant impact on improving the OCR results compared to introducing raw images to the system. Our study highlighted behind-the-seen efforts of sample annotation and selecting the appropriate metadata set and labeling tools to prepare a dataset of document images. We discussed the tools used by researchers and set comparison to discover the most suitable one for annotating an Arabic documents dataset (Aletheia tool). The document image annotation standards were finally created after studying the most common labels and metadata hierarchy needed for representing a document content in many research areas and PAGE format (created by the Aletheia tool) ended up being the most comprehensive annotation scheme for such representation.

Crowdsourcing for datasets annotation

Crowdsourcing has been recently used for constructing different relatively large image, audio and video research datasets through annotation tasks, like segmentation and labeling. It has proved to be very fast and cheap, compared to the expert-based method. However, this has not yet been commonly used in all research areas. Crowdsourcing was not only used for computer vision dataset annotation but also for natural language processing (NLP) datasets. Datasets for named entity recognition [38], image transcriptions [50] were annotated using crowdsourcing. Annotating text corpora, and social media tweets and comments in the form of transcripts, dialects, sentiment analysis and people opinions or orientations were all done through crowdsourcing. Literature about crowdsourcing for annotation of Arabic datasets all lie in this area [3,21,26,29,62,64,65]. However, as far as we know, there is still no attempt for crowdsourcing to annotate scanned documents dataset for the sake of Arabic document image analysis and recognition research, we were the first.

Amazon mechanical Turk (MTurk)

Researchers used the crowdsourcing services offered by Artificial intelligence companies at very small profit like Amazon Mechanical Turk (MTurk) [5], and CrowdFlower (CF) [6]. for the purpose of information collection or fast accomplishment of tedious small tasks. Amazon MTurk might be the first and most popular crowdsourcing platform used by researchers.

---

[5] https://www.mturk.com/mturk/welcome.

[6] https://www.crowdflower.com/ (re-branded as ’Figure Eight’ starting 2018

An MTurk job/HIT The requester divides the entire task to a large number of small jobs (also called human intelligent tasks 'HITs'), that could be done in parallel. Usually the tasks are short time data entry or information extraction, for example answering questions about identifying and/or segmenting an object, or selecting an appropriate label, etc. An instructions set of how the task should be performed with examples of possible instances and common errors are posted to the workers once they accept to do the job. The requester also specifies the maximum time duration for accomplishing the job, and the monetary reward for the given job. The requesters can select workers based on specific qualities related to their tasks and the same HIT could also be assigned to multiple workers for quality assurance. Upon posting the jobs to MTurk, workers try to accept the job, perform the job according to the instructions, and submit it before the specified deadline. Workers might choose long duration jobs with high rewards or a number of short duration jobs with smaller lump-sum. After the jobs submission, the requesters have a deadline to review them and agree to paying/not-paying the workers individually according the job quality. Some requesters choose to offer bonuses beyond the basic reward to some high quality workers as well. Rewards could be as less as two cents and could be as high as tens of dollars according to the job difficulty. The rewards do not represent the entire task pricing, as the budget also include the service provider profit (percentage of the rewards and bonuses). The process is completed by quality assurance procedures and tests to detect spammers and insure agreement between the workers performing the same HIT and also evaluating the annotation accuracy and analyzing errors.

References

1. Abdelaziz, I., Abdou, S.: Altecondb: a large-vocabulary arabic online handwriting recognition database. arXiv:1412.7626 (2014)
2. Dobais, M.A.A, Alrasheed, F.A.G., Latif, G., Alzubaidi, L.: Adoptive thresholding and geometric features based physical layout analysis of scanned arabic books. In: 2018 IEEE 2nd international workshop on arabic and derived script analysis and recognition (ASAR), pp. 171–176. IEEE (2018)
3. Albadi, N., Kurdi, M., Mishra, S.: Are they our brothers? Analysis and detection of religious hate speech in the arabic twittersphere. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 69–76 (2018)
4. Alexey, B., Yao, W.C., Yuan, L.H.: Yolov4: optimal speed and accuracy of object detection. In arXiv:2004.10934 (2020)
5. Almutairi, A., Almashan, M.: Instance segmentation of newspaper elements using mask R-CNN. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pp. 1371–1375. IEEE (2019)
6. Alshameri, A., Abdou, S., Mostafa, K.: A combined algorithm for layout analysis of Arabic document images and text lines extraction. Int. J. Comput. Appl. **49**(23), 30–37 (2012)
7. ALTEC dataset. http://www.altec-center.org/conference/?page_id=87
8. Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome
9. The ASAR Physical Layout Analysis Challenge at the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition, London, U.K., March 2018. https://asar.ieee.tn/competition/
10. 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition, London, U.K., March 2018
11. Asi, A., Cohen, R., Kedem, K., El-Sana, J., Dinstein,I.: A coarse-to-fine approach for layout analysis of ancient manuscripts. In: 14th International Conference on Frontiers in Handwriting Recognition, pp. 140–145 (2014)
12. Barakat, B., Droby, A., Kassis, M., El-Sana, J.: Text line segmentation for challenging handwritten document images using fully convolutional network. In: 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 374–379 (2018)
13. Barakat, B.K., El-Sana, J.: Binarization free layout analysis for arabic historical documents using fully convolutional networks. In: 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 151–155. IEEE (2018)
14. Belaïd, A., Ouwayed, N.: Segmentation of ancient Arabic documents. In: Märgner, V., El Abed, H. (eds.) Guide to OCR for Arabic Scripts, pp. 103–122. Springer, London (2012)
15. Boussellaa, W., Zahour, A., Taconet, B., Alimi, A., Benabdelhafid, A.: PRAAD: preprocessing and analysis tool for Arabic ancient documents. In: 9th International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 1058–1062 (2007)
16. Bukhari, S.S., Azawi, A., Ali, M.I., Shafait, F., Breuel, T.M.: Document image segmentation using discriminative learning over connected components. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, Boston, pp. 183–190 (2010)
17. Bukhari, S.S., Breuel, T.M., Asi, A., El Sana, J.: Layout analysis for arabic historical document images using machine learning. In: 2012 International Conference on Frontiers in Handwriting Recognition, pp. 639–644. IEEE (2012)
18. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: fast and flexible image augmentations. Information **11**(2), 125 (2020)
19. Chen, K., Liu, C.L., Seuret, M., Liwicki, M., Hennebert, J., Ingold, R.: Page segmentation for historical document images based on superpixel classification with unsupervised feature learning. In: 12th IAPR workshop on document analysis systems (DAS), pp. 299–304 (2016)
20. Chen, K., Seuret, M., Liwicki, M., Hennebert, J., Ingold, R.: Page segmentation of historical document images with convolutional autoencoders. In: 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1011–1015 (2015)
21. Cotterell, R., Callison-Burch, C.: A multi-dialect, multi-genre corpus of informal written arabic. In: LREC, pp. 241–245 (2014)
22. Cotterell., Ryan, B., Chris, C..: A multi-dialect, multi-genre corpus of informal written Arabic. In: LREC, pp. 241–245 (2014)
23. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. Comput. Vis. Pattern Recognit. CVPR **2009**, 248–255 (2009)
24. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei, L.F.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
25. Abed, H.E., Märgner, V., Kherallah, M., Alimi, A.M.: ICDAR 2009 online arabic handwriting recognition competition. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 1388–1392. IEEE (2009)
26. El-Mawass, N., Alaboodi, S.: Detecting arabic spammers and content polluters on twitter. In: Sixth International Conference on

Digital Information Processing and Communications (ICDIPC), pp. 53–58 (2016)
27. Elanwar, R., Betke, M.: The ASAR 2018 competition on physical layout analysis of scanned arabic books (PLA-SAB 2018). In: 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 177–182. IEEE (2018)
28. Elanwar, R., Qin, W., Betke, M.: Making scanned arabic documents machine accessible using an ensemble of SVM classifiers. Int. J. Doc. Anal. Recognit. (IJDAR) **21**(1–2), 59–75 (2018)
29. Farra, N., McKeown, K., Habash, N.: Annotating targets of opinions in Arabic using crowdsourcing. In: Second workshop on Arabic natural language processing, pp. 89–98 (2015)
30. Girshick, Ross.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
31. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
32. Hadjar, K., Ingold, R.: Arabic newspaper page segmentation. In: 7th International Conference on Document Analysis and Recognition, pp. 895—899 (2003)
33. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
35. Hesham, A.M., Rashwan, M.A.A., Barhamtoshy, H.M.A., Abdou, S.M., Badr, A.A., Farag, I.: Arabic document layout analysis. Pattern Anal. Appl. **20**(4), 1275–1287 (2017)
36. Kassis, M., El-Sana, J.: Scribble based interactive page layout segmentation using gabor filter. In: 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 13–18 (2016)
37. Ibn Khedher, M., Jmila, H., El-Yacoubi, M.A.: Automatic processing of historical arabic documents: a comprehensive survey. Pattern Recognit. **100**, 107144 (2020)
38. Lawson, N., Eustice, K., Perkowitz, M., Yetisgen-Yildiz, M.: Annotating large email datasets for named entity recognition with Mechanical Turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 71–79 (2010)
39. LabelMe tool. http://labelme.csail.mit.edu/Release3.0/
40. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
41. Mahmoud, S.A., Ahmad, I., Khatib, W.G.A., Alshayeb, M., Parvez, M.T., Märgner, V., Fink, G.A.: KHATT: an open arabic offline handwritten text database. Pattern Recognit. **47**(3), 1096–1112 (2014)
42. Mahmoud, S.A., Luqman, H., Al-Helali, B.M., BinMakhashen, G., Parvez, M.T.: Online-khatt: an open-vocabulary database for arabic online-text processing. Open Cybern. Syst. J. **12**(1), 42–59 (2018)
43. Minghao, L., Yiheng, X., Lei, C., Shaohan, H., Furu, W., Zhoujun, L., Ming, Z.: Docbank: a benchmark dataset for document layout analysis. arXiv:2006.01038 (2020)
44. Neche, C., Belaid, A., Kacem-Echi, A.: Arabic handwritten documents segmentation into text-lines and words using deep learning. In: International Conference on Document Analysis and Recognition Workshops (ICDARW), pp. 19–24 (2019)
45. Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., Papamarkos, N.: Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image Vis. Comput. **28**(4), 590–604 (2010)
46. Pastor-Pellicer, J., Afzal, M.Z., Liwicki, M., Castro-Bleda, M.J.: Complete system for text line extraction using convolutional neural networks and water-shed transform. In: 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 30-35 (2016)
47. Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., Amiri, H.: IFN/ENIT-database of handwritten arabic words. In: Proceedings of CIFED, volume 2, pp. 127–136. Citeseer (2002)
48. Pletschacher S., Antonacopoulos, A.: The PAGE (page analysis and ground-truth elements) format framework. In: 20th International Conference on Pattern Recognition (ICPR), pp. 257–260 (2010)
49. PyTorch sytem of libraries and tools for machine learning. https://pytorch.org/ (2020)
50. Rashtchian, C., Youngand, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using Amazon's mechanical turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 139–147 (2010)
51. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
52. Saad, R.S.M., Elanwar, R., Abdel Kader, N.S., Mashali, S., Betke, M., Asar 2018 layout analysis challenge: using random forests to analyze scanned Arabic books. In: 2nd IEEE International Workshop on Arabic and derived Script Analysis and Recognition (ASAR 2018), London, March 2018, 2018. p. 6
53. Rana S.M.S., Randa I.E., Abdel Kader, N.S., Samia, M., Margrit, B.: BCE-Arabic-v1 dataset: towards interpreting arabic document images for people with visual impairments. In: Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments, pp. 1–8 (2016)
54. Shafait, Faisal, Keysers, D., Breuel, T.: Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 941–954 (2008)
55. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
56. Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J.: A new arabic printed text image database and evaluation protocols. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 946–950. IEEE (2009)
57. Strassel, S.: Linguistic resources for arabic handwriting recognition. In: Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt (2009)
58. Studer, L., Alberti, M., Pondenkandath, V., Goktepey, P., Kolonko, T., Fischeryz, A., Liwicki, M., Ingold, R.: A comprehensive study of imagenet pre-training for historical document image analysis. In: 15th International Conference on Document Analysis and Recognition (ICDAR), pp. 720–725 (2019)
59. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1958–1970 (2008)
60. Wei, H., Seuret, M., Chen, K., Fischer, A., Liwicki, M., Ingold, R.: Selecting autoencoder features for layout analysis of historical documents. In: ACM 3rd International Workshop on Historical Document Imaging and Processing, pp. 55–62 (2015)
61. Wick, C., Puppe, F.: Fully convolutional neural networks for page segmentation of historical document images. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 287–292. IEEE (2018)
62. Wray, S., Mubarak, H., Ali,A.: Best practices for crowdsourcing dialectal arabic speech transcription. In: ANLP Workshop, p. 99 (2015)
63. Wray, S., Mubarak, H., Ali, A.: Best practices for crowdsourcing dialectal arabic speech transcription. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 99–107 (2015)
64. Zaidan, O.F., Callison-Burch, C.: The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In: Proceedings of the 49th Annual Meeting of

the Association for Computational Linguistics: Human Language Technologies: short papers, 2:37–41 (2011)

65. Zaidan, O.F., Callison-Burch, C.: Arabic dialect identification. Comput. Linguist. **40**(1), 171–202 (2014)
66. Zaidan, O.F., Burch, C.C..: The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: short papers-volume 2, pp. 37–41. Association for Computational Linguistics (2011)
67. Zaidan, O.F., Burch, C.C.: Arabic dialect identification. Comput. Linguist. **40**(1), 171–202 (2014)
68. Zhong, X., Jianbin, T., Jimeno, Y.A.: Publaynet: largest dataset ever for document layout analysis. In: 15th International Conference on Document Analysis and Recognition (ICDAR) (2019)