# BCE-Arabic-v1 dataset: Towards interpreting Arabic document images for people with visual impairments

Rana S.M. Saad[1], Randa I. Elanwar[1,2], N.S. Abdel Kader[3] , Samia Mashali[2], and Margrit Betke[2]

[1] Department of Computers and Systems, Electronics Research Institute, Cairo, Egypt
[2] Department of Computer Science, Boston University, USA
[3] Department of Electronics and Communications Engineering, Cairo University, Egypt

rana@eri.sci.eg, randa.elanwar@eri.sci.eg, nemat2000@hotmail.com, samia@eri.sci.eg, and betke@bu.edu

## ABSTRACT

Millions of individuals in the Arab world have significant visual impairments that make it difficult for them to access printed text. Assistive technologies such as scanners and screen readers often fail to turn text into speech because optical character recognition software (OCR) has difficulty to interpret the textual content of Arabic documents. In this paper, we show that the inaccessibility of scanned PDF documents is in large part due to the failure of the OCR engine to understand the layout of an Arabic document. Arabic document layout analysis (DLA) is therefore an urgent research topic, motivated by the goal to provide assistive technology that serves people with visual impairments. We announce the launching of a large annotated dataset of Arabic document images, called BCE-Arabic-v1, to be used as a benchmark for DLA, OCR and text-to-speech research. Our dataset contains 1,833 images of pages scanned from 180 books and represents a variety of page content and layout, in particular, Arabic text in various fonts and sizes, photographs, tables, diagrams, and charts in single or multiple columns. We report the results of a formative study that investigated the performance of state-of-the-art document annotation tools. We found significant differences and limitations in the functionality and labeling speed of these tools, and selected the best-performing tool for annotating our benchmark BCE-Arabic-v1.

## Categories and Subject Descriptors

I.7.5 [**Document Capture**]: Document analysis

## Keywords

Assistive technology for blind users; Arabic document analysis; page layout analysis; optical character recognition (OCR); screen readers; annotation tools; benchmark; training data; image analysis; performance evaluation.



*PETRA '16, June 29-July 01, 2016, Corfu Island, Greece*



ISBN 978-1-4503-4337-4/16/06…$15.00

DOI: http://dx.doi.org/10.1145/2910674.2910725

## 1. INTRODUCTION

To access printed text in their daily routine, tech-savvy individuals without sight use a combination of flatbed scanners and camera-equipped smartphone devices, according to an MIT Media Lab study [29]. They are not satisfied with the word recognition accuracy and processing speed of English OCR software (3 min for digitizing a letter-sized page) and would like cutting-edge tools for reading fragmented text or text on curved surfaces (e.g., a canned goods label). The situation is much direr for individuals with visual impairments in the Arabic-speaking world.

While OCR of digitally-derived text-only pdf files can be considered a solved task for English, this is not at all the case for Arabic [2]. Moreover, the problem of text access for people with visual impairments becomes much more difficult if the text is stored in a file that is not digitally derived, contains non-text elements, or has complex document layouts.

The poor performance of OCR for Arabic compared to Latin-script languages, in particular English, is only one example of a hurdle for Arabic-speaking individuals who are blind. Other natural language processing software such as speech-to-text or text-to-speech conversion and language translation tools also perform at a much lower level for Arabic than for English. One of the goals of this paper is to build awareness in the research community that there is an urgent need for automatic image analysis of Arabic documents.

Arabic is spoken by more than 300 million people, which makes it the 5th-most-spoken language worldwide. Historically, it has had an important role in preserving the flow of knowledge between cultures and eras. Because Arabic characters are used to represent other languages, developing image analysis tools that interpret Arabic script may not only serve individuals in the Arab world, but also individuals who speak Urdu, Persian, Pashto, Kurdish, Jawi, Wolof, Pular, Amharic, Hausa, Swahili, Tigrigna, or Berber.

The Arabic content on the internet was estimated to include 2 billion pages in 2012 [1]. The Arab world is trying to increase the Arabic content on the internet by digitizing and archiving both contemporary data as well as documentation of ancient heritage, for example, the Arabic Collection Online [4] and the Islamic Heritage Project [15]. Making Arabic documents accessible to computer users with or without visual impairments requires image-analysis tools that can interpret the digital images of these documents. "The objective of document image analysis is to recognize the text and graphics components in images of documents, and to

extract the intended information as a human would" [16]. Document image analysis tools are needed for mobile products, for example, FingerReader [29], the Text Detective by Blindsight [32], the EyePal scanner and reader [9], and the assistive eyeglasses by OrCam [20].

The development and evaluation of intelligent systems for image analysis of documents require a large number of document samples with annotated ground truth. The ground truthing process for complex documents in any language is still done manually or semi-automatically. It is an expensive, time-consuming, and typically application-dependent process.

The lack of satisfactory solutions for text-access has resulted in proliferation of temporary alternatives based on human collaboration to receive quick and no-cost help in everyday situations. Offline help, for example, shows up in individual and group volunteerism for audio books recording. Online help shows up in crowd collaboration for visual question answering [17]. However, these solutions cannot answer all text-accessibility needs that arise. Individuals who are visually impaired may want to access a printed book or newspaper, they may not have internet coverage at an affordable cost, they may not have the appropriate technical skills to use online collaboration, and, on top of all this, their language might not be supported.

In this paper, in order to facilitate research on automated image analysis of Arabic documents, we introduce a large database of Arabic-document images and their annotations, called BCE-Arabic-v1, located at `http://www.cs.bu.edu/faculty/betke/BCE`. BCE-v1 stands for the first benchmarking effort by team members from **B**oston University, **C**airo University, and **E**lectronics Research Institute. To the best of our knowledge, this is the first large dataset that provides a representative variety of document content, including text and non-text elements, for document layout analysis (DLA) for the Arabic language. Specifically, we make BCE-Arabic-v1 available as a training and performance-evaluation tool for development of machine learning systems that analyze Arabic documents with normal and complex layouts. We also investigate the limitations of annotation tools that are currently available for manual ground truthing. The contributions in this paper can be summarized as

1. Creating awareness about the inaccessibility of document images of Arabic script to individuals with visual impairments;
2. Showing that Arabic documents become more accessible if the page layout is provided to the OCR engine;
3. Providing BCE-Arabic-v1, a benchmark dataset of images of Arabic documents with ground-truth annotations of their page layouts;
4. Surveying the state-of-the-art ground-truthing tools available for document image annotation, irrespective of language;
5. Evaluating the performance of these tools when applied to a variety Arabic document layouts and investigating their strengths and limitations based on multiple metrics.

## 2. PDF INACCESSIBILITY

The portable document file format PDF is popular because of its ability to stay true to the intended display irrespective of the document reader or operating system used. PDF tags provide a hidden structured representation of the PDF content that can be presented to screen readers used by people with visual impairments. PDF tags exist for accessibility purposes only and have no visible effect on the PDF file [21].

PDF files can be categorized according to their tags [2]. The first category includes formatted text and graphics PDF files, which are fully tagged and have layout and text information available. The second category consists of searchable-image PDF files, which are scanned copies associated with a hidden text layer for accessibility but no information about the layout. The third category includes raster-image PDF files, which are scanned copies of documents that do not contain any tags.

The first and second categories of PDF files are deemed "accessible" especially if they contain Latin script. It is important to note though that assistive screen readers tested on fully-tagged digitally-born PDF files generated from MS Word documents do not perform the same for Arabic as they do for English according to a performance evaluation study [2]. While this study was an important first step in pointing out the inaccessibility of digitally-born PDF files containing Arabic script, its experiments were conducted on an extremely limited dataset. Only 3 documents were created and tested. This exemplifies the extent to which researchers lack appropriate datasets to investigate the problem of text access for people with visual impairments who speak a language that uses Arabic script.

As we show in the next section, the third category of PDF documents is the most problematic. The content of raster-image PDF files of Arabic text is currently inaccessible to users with visual impairments.

## 3. A PILOT STUDY OF ARABIC OCR WITH AND WITHOUT DLA

We conducted a pilot study to assess the efficacy of OCR software when applied to raster-image PDF files of Arabic documents. We used Tesseract an Open Source OCR engine (https://github.com/tesseract-ocr/tesseract/wiki), which is widely used by researchers and programmers, as it supports a variety of languages, including Arabic. Tesseract has an application programming interface (API) for building large software systems that use Tesseract as a front or back end.

In the first round of our pilot study, we applied Tesseract to scanned book pages with the goal to measure whether Tesseract can recognize their textual contents (Figure 1 left). We tested a variety of page layouts and found that the OCR software had difficulty interpreting the documents that have page layouts with several textual and non-textual components (Figure 1 middle).

We next considered how DLA and OCR software successfully work together in interpreting the textual contents of an English image document: DLA software first analyzes the page layout of the document and identifies the location and type of the text and graphics components of the document. The text blocks are then interpreted by the OCR engine in an appropriate order. To establish this order, a hierarchical structure of document components has to be created by the DLA system (for example, in HTML, XML, or CSS format). This task is difficult for highly complex layouts that include text and non-textual components, such as photographs, charts, diagrams, specially-formatted text

like tables, and organization elements like borders and separators. Some DLA software for English documents also provides image quality enhancement and noise elimination tools that are designed to supply the OCR engine with easier-to-interpret input images.

In the second round of our pilot study, we wanted to find out if Tesseract could recognize the textual content of documents with complex layouts as long as the location and type of the layout components had been identified. We simulated the input that a DLA system would provide to Tesseract by using the ground-truth annotations of the text components of the images. Specifically, we supplied Tesseract with subimages of the text components of the documents in the appropriate order. (In some cases, we had to enlarge small subimages of headings because Tesseract expects certain input image dimensions.) With DLA as a pre-processing step, Tesseract was able to interpret a significantly larger portion of the Arabic text (Figure 1 right) compared to our text without DLA.

Our pilot study could be extended to other OCR packages or websites that support the Arabic language. While the resulting text interpretations may somewhat differ from those provided by Tesseract, they are likely more accurate with DLA than without.

The results of our study suggest that the inaccessibility of Arabic image documents is in large part due to the failure of the OCR engine to understand their page layout. Automated layout recognition systems of Arabic documents are therefore much needed. They would likely boost the efficacy of Arabic OCR and thus have an important quality-of-life impact on people with visual impairments who are dependent on assistive technology to access such documents.

## 4. EXISTING BENCHMARKS FOR DLA RESEARCH ARE SMALL

The availability of publicly-available datasets is crucial for accelerating research on automated layout recognition systems of Arabic documents. If the datasets have a sufficiently large number and variety of documents, they can be used as benchmarks for comparing the performance of research systems. Currently available datasets for DLA research, however, are small.

The dataset provided by Bukhari et al. [5] contains 25 images from books and newspapers, including multi-script images that contain both English and Arabic script; the Hadjar and Ingold datasets [11, 12, 13] contain between 50 to 150 pages from three different newspapers (Annahar, AL Hayat, and AL Quds), and the dataset by ElShameri et al. [3] contains 200 pages from newspapers. The database by the Environmental Research Institute of Michigan [26] consists of 750 images of pages from machine-printed Arabic books and magazines.

DLA datasets for Spanish are also small. The Spanish handwritten historical documents public dataset GERMANA [23] consists of 764 pages, and the RODRIGO dataset [27] of 853 pages. DLA datasets for Chinese are somewhat larger: the SentiCorp dataset [31] contains 1,021 documents; the CASIA dataset for offline handwriting [19] has 5,090 pages. In stark contrast, for English, enormous efforts in collecting and annotating DLA data have been undertaken, involving entire books.

There are datasets to support Arabic OCR research: The

**Figure 1: Pilot study evaluating Arabic OCR performance with and without DLA. Sample images with ground-truth text regions (left) were interpreted by the Tesseract OCR engine without DLA (middle) and with DLA (right) support. Without DLA, Tesseract could not recognize most words (black). With DLA, Tesseract understood most words correctly (red) or with at most 2 character errors (green). Occasionally Tesseract became stuck interpreting a word (blue). These results suggest that the inaccessibility of Arabic image documents is in large part due to the failure of the OCR engine to understand the page layout of the documents.**

MADCAT dataset (Multilingual Automatic Document Classification Analysis and Translation) contains 38,000 handwritten Arabic pages [30]; the IFN/ENIT-database contains 2,200 handwritten forms from 411 writers and about 26,000 binary-word images by the Institute of Communications Technology (IFN) of the Technical University Braunschweig, Germany, and the Ecole Nationale d'Ingenieurs de Tunis (ENIT), Tunisia [22].

## 5. BCE-ARABIC-V1 BENCHMARK

The purpose of the BCE-Arabic-v1 dataset is to (1) accelerate the development of automated solutions for a variety of DLA problems for Arabic document images and (2) help with benchmarking and comparative evaluation of research efforts.

In the first stage, we have collected 1,833 images of pages with different layouts from 180 books produced by the same

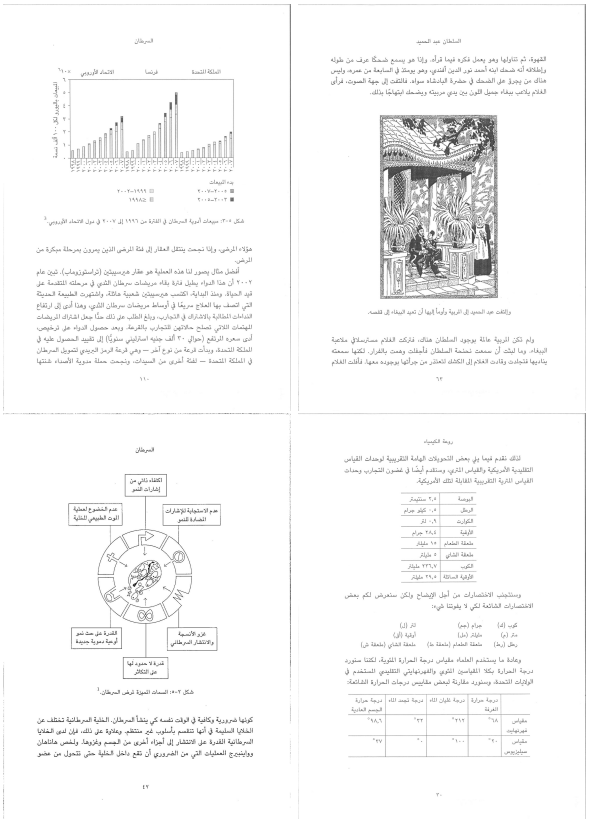

**Figure 2: Example documents in BCE-Arabic-v1, groups b and c.**

publisher Hindawi. Examples are shown in Figure 2. Every page contains Arabic script. The images are scanned to 400 dpi resolution and stored in raster-image PDF format. The BCE-Arabic v1 database contains the following layout types:

a) 1,235 images containing only text; its components are titles, page headers, body text, footers, footnotes, captions in various font sizes and with a range of formats, and

b) 383 pages with text and images,

c) 179 pages with text and graphic elements (charts and diagrams),

d) 24 pages with text and tables,

e) 29 images with text in mixed single & double columns.

It is noteworthy to discuss why we created BCE-Arabic-v1 by scanning the documents ourselves instead of downloading document images from the internet and simply ground-truthing their page layouts. After all, a large number of Arabic documents are available in form of scanned books and journals. Our reasons are as follows:

- The layout variability of web documents is limited.
- The document images are of low-quality due to low-resolution B&W scanning intended to minimize the upload file size.
- Most of the books are copyrighted and have access restrictions (e.g. viewable but not downloadable).
- Many images contain digital library watermarks, added by publishers or libraries.
- It is not easy to discern if the copyrights have been infringed by the upload.
- Most uploaded Arabic image content is of religious nature and uses a script with a multitude of diacritical marks (i.e., small glyphs used as phonetic guides). Documents on science, literature, or art do not use diacritics and are therefore easier to annotate.

# 6. LAYOUT ANNOTATION TOOLS FOR ARABIC DOCUMENTS

To find an appropriate tool for annotating the layout of the document images in our BCE-Arabic-v1 dataset, we conducted a comparison study of five state-of-the-art layout annotation tools – Pixlabeler [25] DIVADIA [6], GEDI [8], TrueViz [18], and Aletheia [7] – and the general-purpose tool Microsoft Paint. We investigated the quality of the annotation labels that could be obtained with each tool and measured how much human time and effort was required by each tool.

Effective ground-truthing tools enable the user to determine regions of interest (ROIs), also called document zones, and to annotate these ROIs according to a defined set of labels (metadata or tags). The labels are then stored in a hierarchical structure, usually in XML format, and serve as the ground-truth reference. The outline of an ROI can be a bounding box or arbitrarily-shaped polygon that the user defines with click-and-drag mouse operations. The mandatory metadata needed for each ROI is its location in the image and its type. Optional metadata include a unique page ID, unique ROI ID, ROI classification as text or non-text, ROI logical role attributes (header, footer, text body, page number, caption, heading, images, tables, charts, bars, and logos), page font attributes (type, style, size), ROI language, the ASCII representation of text ROIs, text reading direction, the number of segmented ROIs, and reading order of the ROIs.

We carefully designed a sample set from our BCE-Arabic dataset that focused on variety of content. We selected 25 images, 5 per group, covering the variety of layouts that exist in BCE-Arabic dataset:

A: normal layout text, including different font sizes (e.g., titles, headers, footers, page numbers),

B: normal layout text and photographs,

C: normal layout text, photographs, and graphic elements (tables, charts, equations, or text in a frame), covering 1/3 of the image.

D: complex layout (multi column) text with different font sizes (titles, headers, footers and/or page numbers),

E: complex layout (multi column) text and photographs.

We considered a document image fully annotated if each text and non-text image component was outlined and labeled with metadata. In our study, we compared the tools with respect to several metrics, including (1) their support for manual segmentation of image regions of interest, (2) their support for a metadata annotation, (3) the time consumed in annotation, (4) the resulting output format, and (5) the ease-of-use of the tool. The performance of the six tested ground-truthing tools is summarized in Tables 1–3. The performance metrics were averaged for the 5 documents in each of the 5 groups.

**Figure 3: Pixlabeler regions by color**


**Figure 4: Divadia output examples. Top: Ground-truth annotation of a Parzival document (used with permission) Bottom: XML file.**

## 6.1 Microsoft Paint

MS Paint is a general tool for editing images and as such does not have special provisions for storing metadata for document analysis. Nonetheless, it has been used successfully for ground-truthing document images [14, 28] where the pixels in each region of interest are assigned the same color, and the pixels in different regions different colors. Regions are segmented manually with polygon-shaped borders and labeled with the zone type, text or non-text.

## 6.2 Pixlabeler

Pixlabeler (V2.1) [25] provides more ground-truthing controls than MS Paint. For example, it can detect text lines automatically. It can also automatically load and compare multiple labeling results for a given image that have been produced by other users or an automated process.

Ground truthing happens in one of two modes. In the 'segmentation mode,' with each new zone selection, a new color label (among 65,534 colors) is selected automatically. In the 'region mode,' the ROI can be labeled with one of the built-in labels which are machine-printed text, handwritten text, handwritten graphics, stamps, or salt-and-pepper noise.

Pixlabeler produces an image in PNG format (see Figure 3, and an XML file that describes the image attributes such as its location path, width and height, label data, and edit mode. We found it inconvenient that the XML file neither contains any information about the bounding box coordinates of the colored labeled zones nor any of the entered ROI labels.

## 6.3 DIVADIA

DIVADIA was created to enable the annotation of historical documents [6]. ROIs can be labeled only as page, text block, text line, decoration, and comment (Figure 4). Decorative elements include capitals, decorative initials, and ornaments. Ground truthing is achieved by drawing arbitrary shaped polygons around regions of interest, then representing the metadata in a XML output file.

From our experience using DIVADIA we found: The polygon vertices can be relocated to adjust the required zone, however it is difficult to edit a drawn rectangle and we found it was easier to delete it and redraw the region. Keyboard shortcuts are not supported, for example, CTRL+A selects all documents instead of ROIs. To delete an ROI, the user has to select one of its corner points and press the delete button on the GUI. Zooming in and out is available.

We found that five zone labels were too few for us to be able to annotate our sample documents satisfactorily. DIVADIA does not have labels to annotate photographs, tables, or charts. Moreover, since DIVADIA is only designed to link between the decoration ROIs and text lines and not to provide guidance to assistive screen readers, it does not have the capability to store the order of zones. It is not clear if DIVADIA supports Unicode text for text entry of the type 'comment.' When we entered Arabic characters, meaningless symbols appeared in the output XML.

## 6.4 GEDI

GEDI (v2.4) stands for **G**roundtruthing **E**nvironment for **D**ocument **I**mages [8]. It enables manual segmentation of zones using rectangular or arbitrarily-sized polygons on binary or grayscale images. At the beginning of the project configuration, mandatory and optional attributes can be set for each zone and displayed for future use, including zone types, colors, keys, visibility for all documents, and the possible attribute values. The zone ID attribute is used to define the reading order for assistive screen readers. GEDI also offers loading and displaying specific images and metadata from sources other than manual data entry, commenting and tracking, automatic connected components (CC) detection and automatic zoning using run-length encoding. The default output of GEDI is in XML format but a BMP image can be obtained as well (Fig. 5). The XML file contains image ID, width, height, and then, for each region, ID, next zone ID, bounding box (BB) coordinates, and zoning level.

Although GEDI offers run length encoding for automatic zoning, this feature did not work in our dataset. We found it only worked with a degraded (noisy) version of our data. GEDI supports Unicode encoded text in numerous languages, including Arabic.

## 6.5 TrueViz

TrueViz [18] is a Java tool with an interface for zoning and annotating data in the DAFS XML standard [10]. TrueViz (v1.0) supports only the tiff format as input. ROIs are segmented with rectangular or arbitrarily-shaped polygons and then labeled using a side menu to enter the metadata of the page or ROIs. The available page metadata are page


**Figure 5: Gedi image regions.**

ID, type, number, columns, page next, page zones, language, font, alignment, reading direction, classification, and ground truth (GT) text.

The ROI metadata have different levels. For the character level, the types are ID, bounding box (BB) corner coordinates, next character, and GT text. The types for the word level are the same as for the character level, plus the number of characters in the word. For the line level the types are ID, BB corner coordinates, next line, number of characters, and GT text. The types for the paragraph level are ID, BB corner coordinates, next, font, alignment, reading direction, classification, and GT text. All these fields are to be entered manually.

Segmentation of ROIs is done one-by-one using the main menu with no editing facility to correct errors. ROI bounding-box coordinates are computed automatically. Any missing annotation does not appear in the output XML. TrueViz supports Unicode-encoded text in Arabic, Chinese, English, Japanese, Korean, and Russian by providing a software keyboard for text ground-truth entry.

## 6.6 Aletheia

Aletheia [7] is part of a complete performance analysis infrastructure created and maintained by PRima research center (University of Salford, Manchester,UK). This infrastructure encompasses XML formats, ground-truthing and evaluation tools, validators, converters, and viewers. Alethia (v2.2) allows defining and labeling of zones, connecting them in a certain logical order, and text entry for the evaluation phase of the OCR system.

The output of Aletheia is saved in an XML file in the PAGE (Page Analysis and Ground truth Elements) format framework [24]. The PAGE format is a comprehensive, widely-used format in document analysis. For example, it has been used to represent the ground truth of the datasets in the ICDAR page segmentation competition series and the ICFHR 2014 Handwritten Text Recognition on the tranScriptorium Dataset competition [6]. Aletheia does not an image format as an output.

The interface of Aletheia is built with a Multi Document Interface (MDI) that facilitates the switch between different modes of operations: region, zone, text line, word, and glyph modes. It offers image preprocessing options like binarization, border removal, and noise reduction.

Automatic top-down and bottom-up region detection methods are available. Aletheia also contains a manual correction tool, page auto-analysis, and output structure editing. The set of metadata attributes includes region types, region ID, reading direction, language, and font type. The logical relation between the page components is provided. The tool also includes a validator which points out the missing annotations in the layout regions. Aletheia has an integrated OCR engine as part of facilitating the layout ground-truth production.

Binarization is necessary to access some features of Aletheia (v2.2), so we used the Otsu binarization method and saved the resulting image in .tiff format. We performed segmentation semi-automatically with a "smearing feature" (200 smearing threshold with 100 smearing increment) and the "fine contour rectangle" option. Fitted contours of the ROIs can be generated and converted to sharp rectangular shape. The reading order of zones for screen readers is determined automatically. So, the annotation time consumed is dedicated to labeling ROIs.

Although Aletheia offers automatic page analysis by running Tecceract OCR internally, it did not work with the Arabic script in the test dataset. The reason could be that the API automatic language detection option is either off or failed to recognize the fonts. However, Alethia does support Unicode-encoded text in all languages, including Arabic, and special non-Unicode symbols.

# 7. DISCUSSION OF RESULTS

Our results indicate that GEDI, Aletheia, and TrueViz have many common features for annotation and zoning that make them the preferable tools for ground truthing of documents. Two important factors should be considered by the user when selecting which tool to use: time and tool flexibility.

Regarding tool flexibility, we note that Aletheia (v2.2) behaves the same as GEDI (v2.4) for manipulating the document with ROIs Editing and short-keys facilities. TrueViz, however, gives no aid to accelerate the zoning by any short keys.

The average annotation time for each tool and each document group is listed in Table 3. It has to be noted that Groups A, B, and C have on average around 4, 6, and 5 zones per document page respectively. Group D has on average 11 zones and group E around 22 zones. The ground-truthing time is directly proportional to the number of zones. The same annotator was asked to test all tools after having trained on each to be familiar with the tool options.

The time comparison reveals that TrueViz is very slow compared to GEDI and Aletheia. Too much time was spent to manually annotate all attributes for each zone.

Zoning in GEDI is manual (smearing works with noisy images only) while in Aletheia it is semi-automatic (smearing works). The annotation in Aletheia is performed page-by-page while GEDI is performed once for all documents and the different field values only are changed.

Considering the overall annotation time, Aletheia was a more efficient tool than GEDI, and that is the reason we selected it for annotating our dataset. Aletheia also supports a rich set of labels of its PAGE XML output file.

The layout ground truthing of the current version of the BCE-Arabic benchmark (v1) is complete. Text ground truthing

**Table 1: Zoning Properties of the Tools**

| | Annotated Unit | Smallest Annotated Zone | Editing Zoned ROIs | Zooming | Keyboard Shortcuts | Rectangular Shape | Isometric Polygon | Arbitrary Polygon | Image Pre-Processing |
|---|---|---|---|---|---|---|---|---|---|
| MS-Paint | Pixel | Word | Yes | Yes | Enabled | Yes | No | Yes | No |
| Pixelabeler | Pixel | Word | No | Yes | Disabled | Yes | No | No | No |
| DIVADIA | Pixel | Text line | Yes | Yes | Disabled | Yes | Yes | No | No |
| GEDI | CC | Paragraph | Yes | Yes | Enabled | Yes | Yes | No | No |
| Althethia | CC | Character | Yes | Yes | Enabled | Yes | Yes | Yes | Yes |
| TrueViz | Pixel | Character | No | No | Disabled | Yes | No | Yes | No |

**Table 2: Labeling Capabilities of the Tools**

| | Output | | Metadata Entry Enabled | Attribute Definition/ Data Entry | Mandatory attributes | | | ROI type | Reading Order Tracking | Textual Content Entry | Annotation Validation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Image | XML | | | GT language | Reading Order 'Page & ROI ID' | BB Coordinates | | | | |
| MS-Paint | Yes | No | No | None | None | No | No | No | n/a | No | No |
| Pixelabeler | Yes | Yes | Yes | None | None | No | No | Yes | n/a | No | No |
| DIVADIA | Yes | Yes | Yes | None | Unknown | No | Yes | Yes | n/a | No | No |
| GEDI | Yes | Yes | Yes | Manual | Arabic supported | Yes | Yes | Yes | Manual | Yes | No |
| Althethia | No | Yes | Yes | Drop menus | Arabic supported | Yes | Yes | Yes | Manual & Automatic | Yes | Yes |
| TrueViz | No | Yes | Yes | Manual | Arabic supported | Yes | Yes | Yes | Manual & Automatic | Yes | No |

**Table 3: Average Annotation Time (in min)**

| | Layout Group | | | | |
|---|---|---|---|---|---|
| Tool Name | A | B | C | D | E |
| MS-Paint | 0:38 | 1:02 | 0:40 | 1:29 | 5:08 |
| Pixelabeler | 0:19 | 1:24 | 0:22 | 0:32 | 2:04 |
| DIVADIA | 0:19 | 0:31 | 0:10 | 0:12 | 2:28 |
| GEDI | 1:25 | 1:58 | 1:28 | 1:57 | 6:32 |
| Alethia | 1:17 | 1:25 | 1:28 | 1:38 | 3:54 |
| TrueViz | 6:18 | 9:09 | 8:25 | 11:19 | 24:54 |

will follow. A new version with larger variance of layouts from different book publishers is in progress. Successive stages of collection and annotation will follow involving different themes of the scanned material.

# 8. CONCLUSIONS

In this paper, a case was made for the urgent need for text-access tools that give individuals with visual impairments independence and privacy. A particular focus was the analysis of Arabic documents. The success of numerous tasks is based upon the success of layout analysis of Arabic documents. Tasks include improving Arabic OCR, preserving historical heritage with document digitization and archiving, and enabling users with and without visual impairments to search and retrieve information in Arabic image documents.

BCE-Arabic is an ongoing project in which the 1st stage has been completed by collecting over 1,800 images of Arabic book pages with significant layout and page content variability. We have annotated the dataset with the document analysis tool Aletheia [7], the winning tool in our formative study in which we compared six state-of-the-art annotation tools. The second stage of the BCE-Arabic project is in progress and will extend the work described in this paper. We will add a substantial amount of data to BCE-Arabic and consider crowdsourcing for ground-truthing.

We hope that the current and future versions of BCE-Arabic will serve the community of researchers as training and benchmarking data for machine learning systems so that timely document-analysis solutions that assist people with visual impairments can be built.

We share our image dataset and annotations, BCE-Arabic-v1, with the research community to support application and future extensions of this work (http://www.cs.bu.edu/faculty betke/BCE).

# ACKNOWLEDGMENTS

The work was partially funded by the National Science Foundation, grant 1337866 (to M.B.) and the Cairo Initiative Scholarship Program (to R.E.).

# 9. REFERENCES

[1] A. Alarifi, M. Alghamdi, M. Zarour, B. Aloqail, H. Alraqibah, K. Alsadhan, and L. Alkwai. Estimating the size of Arabic indexed web content. *Scientific Research and Essays*, 7(28):2472–2483, July 2012.

[2] A. M. AlMasoud and H. S. Al-Khalifa. Investigating accessibility problems of Arabic PDF documents. In *Fourth IEEE International Conference on Information and Communication Technology and Accessibility (ICTA)*, 2013.

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

[4] Arabic Collections Online, New Year University. http://dlib.nyu.edu/aco, 2016.

[5] S. Bukhari, F. Shafait, and T. M. Breuel. High performance layout analysis of Arabic and Urdu

document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1275–1279, Sept. 2011.

[6] K. Chen, M. Seuret, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold. Ground truth model, tool, and dataset for layout analysis of historical documents. In *Proc. SPIE 9402, Document Recognition and Retrieval XXII*, Feb. 2015.

[7] C. Clausner, S. Pletschacher, and A. Antonacopoulos. Aletheia – an advanced document layout and text ground-truthing system for production environments. In *IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 48–52, Sept. 2011.

[8] D. Doermann, E. Zotkina, and H. Li. GEDI – a GroundTruthing Environment for Document Images. In *Ninth IAPR International Workshop on Document Analysis Systems*, June 2010. http://lampsrv02.umiacs.ugmd.edu/projdb/project.php?id=53.

[9] Eye-Pal ROL portable scanner and reader, blindness solutions by FreedomScientific. http://freedom-scientific.com/Products/Blindness, 2016.

[10] T. Fruchterman. DAFS: A standard for document and image understanding. In *Proceedings of Symposium on Document Image Understanding Technology*, pages 94–100, Oct. 1995.

[11] K. Hadjar and R. Ingold. Arabic newspaper page segmentation. In *International Conference on Document Analysis and Recognition (ICDAR)*, Aug. 2003.

[12] K. Hadjar and R. Ingold. Physical layout analysis of complex structured arabic documents using artificial neural nets. In *Document Analysis Systems VI*, pages 170–178, 2004.

[13] K. Hadjar and R. Ingold. Logical labeling of Arabic newspapers using artificial neural nets. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 426–430, Aug. 2005.

[14] S. M. Hanif and L. Prevost. Texture based text detection in natural scene images – a help to blind and visually impaired persons. In *Conference on Assistive Technologies for People with Vision & Hearing Impairments*, Aug. 2007.

[15] Islamic Heritage Project, Harvard University. http://ocp.hul.harvard.edu/ihp/scope.html, 2016.

[16] R. Kasturi, L. O'Gorman, and V. Govindaraju. Document image analysis: A primer. *Sandhana*, 27(1):3–22, 2002.

[17] W. S. Lasecki, P. Thiha, Y. Zhong, E. Brady, and J. P. Bigham. Answering visual questions with conversational crowd assistants. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 18, 2013.

[18] C. H. Lee and T. Kanungo. The architecture of TrueViz: a groundTRUth/metadata editing and VisualiZing tool. *Pattern Recognition*, 36(3), 2003. http://www.kanungo.com/software/software.html#trueviz.

[19] C. Liu, F. Yin, D. Wang, and Q. Wang. CASIA online and offline Chinese handwriting databases. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 37–41, Sept. 2011.

[20] OrCam-MyEye, wearable device with a smart camera designed to assist people who are visually impaired. http://www.orcam.com, 2016.

[21] Pdf accessibility. http://webaim.org/techniques/acrobat, 2016.

[22] M. Pechwitz, S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri. IFN/ENIT-database of handwritten Arabic words. In *Colloque International francophone sur l'ecrit et le document (CIFED), Hammamet, Tunisie*, pages 127–136, Oct. 2002.

[23] D. Perez, L. Tarazon, S. N., C. F., O. Ramos Terrades, and J. A. The GERMANA database. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 301–305, 2009.

[24] S. Pletschacher and A. Antonacopoulos. The PAGE (Page Analysis and Ground-Truth Elements) format framework. In *20th International Conference on Pattern Recognition (ICPR)*, pages 257–260, 2010.

[25] E. Saund, J. Lin, and P. Sarkar. Pixlabeler: User interface for pixel-level labeling of elements in document images. In *10th IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 646–650, July 2009.

[26] S. Schlosser. ERIM Arabic database. document processing research program, information and materials applications laboratory. Technical report, Environmental Research Institute of Michigan, 1995.

[27] N. Serrano, F. Castro, and A. Juan. The RODRIGO database. In *International Conference on Language Resources*, pages 2709–2712, May 2010.

[28] F. Shafait. *Geometric Layout Analysis of scanned documents*. PhD thesis, Technical University Kaiserslautern, 2008.

[29] R. Shilkrot, J. Huber, C. Liu, P. Maes, and S. C. Nanayakkara. FingerReader: a wearable device to support text reading on the go. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 2359–2364, 2014.

[30] S. Strassel. Linguistic resources for Arabic handwriting recognition. In *The Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt*, Apr. 2009.

[31] S. Tan and J. Zhang. An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications*, 34(4):2622–2629, 2008.

[32] Text Detective by Blindsight, an app for the iPhone and Android that can detect text and read it out aloud. http://blindsight.com/textdetective, 2016.