International Journal on Document Analysis and Recognition (IJDAR) (2018) 21:59–75
https://doi.org/10.1007/s10032-018-0298-x

ORIGINAL PAPER

Making scanned Arabic documents machine accessible using an ensemble of SVM classifiers

Randa Elanwar[1] · Wenda Qin[2] · Margrit Betke[2]

Received: 4 January 2017 / Revised: 23 January 2018 / Accepted: 23 March 2018 / Published online: 2 April 2018

Abstract

Raster-image PDF files originating from scanning or photographing paper documents are inaccessible to both text search engines and screen readers that people with visual impairments use. We here focus on the relatively less-researched problem of converting raster-image files with Arabic script into machine-accessible documents. Our method, called ECDP for "Ensemble-based classification of document patches," segments the physical layout of the document, classifies image patches as containing text or graphics, assembles homogeneous document regions, and passes the text to an optical character recognition engine to convert into natural language. Classification is based on the majority voting of an ensemble of support vector machines. When tested on the dataset BCE-Arabic [Saad et al. in: ACM 9th annual international conference on pervasive technologies related to assistive environments (PETRA'16), Corfu, 2016], ECDP yielded an average patch classification accuracy of 97.3% and average F_1 score of 95.26% for text patches and efficiently extracted text zones in both paragraphs and text-embedded graphics, even if the text is rotated by 90° or is in English. ECDP outperforms a classical layout analysis method (RLSA) and a state-of-the-art commercial product (RDI-CleverPage) on this dataset and maintains a relatively high level of performance on document images drawn from two other datasets (Hesham et al. in Pattern Anal Appl 20:1275–1287, 2017; Proprietary Dataset of 109 Arabic Documents. http://www.rdi-eg.com). The results suggest that the proposed method has the potential to generalize well to the analysis of documents with a broad range of content.

Keywords Arabic document analysis · Physical layout analysis · Page layout analysis · Optical character recognition (OCR) · Screen readers · Classifier ensemble · Page zone classification · Creation of structured meta data

1 Introduction

In our digital world, the pervasive file format for data transfer and sharing is the portable document format (PDF). It is supported by prevailing operating systems and application programs and enables us to search for text within a digitally derived ("born-digital") document. We can add navigation tags to make digitally derived documents specifically accessible for people with disabilities. However, if the document was not digitally derived but resulted from scanning or photographing a hard copy of a paper document, its contents are merely stored in a raster graphics PDF file without searchable text streams or embedded navigation tags.

The problem of converting a raster-image PDF file into a machine-accessible PDF document that facilitates text search and supports navigational assistance to PDF readers for people with visual impairments has not been solved in general. Commercial solutions are provided by optical character recognition (OCR) software for the special case of interpreting document images that contain text in machine-created script in simple page layouts. Currently available OCR software interprets text in the Latin alphabet relatively reliably but does not solve the general conversion problem for a variety of reasons:

(1) The original hard copy may be of poor quality (e.g., contain ink bleeding or may be ripped), (2) the document imaging process may yield text in insufficient resolution or may have introduced skew or extraneous lines, (3) the OCR engine may not be able to handle certain fonts, printed text in non-Latin script, decorative drop caps, or handwritten text, (4) the document may contain various non-text elements, such as diagrams and images, and (5) the layout of

✉ Randa Elanwar
eng_R_I_Elanwar@yahoo.com

[1] Electronics Research Institute, Cairo, Egypt

[2] Boston University, Boston, USA

the document elements may be complex, including headlines, footnotes, tables, and lists. The document layout is particularly complex if it contains both text elements and illustrations, a non-uniform background, text regions with different orientations, non-rectangular regions, curved text, or more than one text column.

Our contribution, described in this paper, is a supervised machine learning method for the document layout analysis (DLA) that may be incorporated into an end-to-end system pipeline for making raster-image PDF files accessible. The literature on DLA (e.g., [1]) distinguishes *physical layout analysis* (PLA), also called *geometric layout analysis*, and *logical layout analysis* (LLA). 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. An end-to-end system for converting document images into accessible PDF files may include both types of layout analysis. The PLA component interprets the raw page content and identifies zone types, whereas the LLA component provides descriptive tags and navigational aids needed for fully accessible PDF files. Our work provides a solution for PLA, segmenting layout zones, so that the text in the document can be made accessible with an OCR engine.

Research on document layout analysis has mostly focused on English text. Exceptions include documents in medieval German and Latin [2], as well as Arabic [3–5]. Analysis of documents in Arabic are also the focus of our work. Arabic is spoken by more than 300 million people world-wide. Developing methods to analyze documents in Arabic script serves not only the Arab world, but also cultures in Asia (Urdu, Persian, Pashto, Kurdish) and North Africa (Swahili, Berger, Hausa) whose languages use Arabic characters.

While language-independent solutions are desirable, existing DLA methods have been developed for documents in a particular language or language group. This way, the automatic interpretation of the document can take into account the printed characters or handwritten scripts, symbols, text layouts, reading order (left-to-right, right-to-left, or vertical), etc., of the specific language.

Leveraging language-specific features for DLA requires a sufficient supply of training data in that language, especially if the DLA system uses supervised machine learning. The dearth of large datasets of Arabic documents and their layout annotations to train DLA systems was recently described by Saad et al. [6], who, as a result of their study, were motivated to collect and introduce a new, publicly available dataset, BCE-Arabic [7]. In our work, we use BCE-Arabic, as well as two proprietary datasets collected by others [8,9].

1.1 Problem definition

In summary, the problem that we solve here is to design a method that takes as input a raster image of an Arabic document, which may contain text and graphic elements, and produces as output an XML file that contains the location and type information of every homogeneous zone in the document. The method must enable an OCR engine to access the text stream in each text zone of the document, thus making the text in the previously inaccessible PDF file now accessible.

2 Related work

2.1 Layout segmentation

Typically, the first step of physical layout analysis is the task of segmenting the document image into homogeneous regions. This is usually performed using one of two approaches: (1) the 'top-down approach' initially considers the page to form one huge zone and then conducts a number of successive divisions until no zone is left that could be further segmented into homogeneous zones, and (2) the 'bottom-up approach' that starts at the smallest element level (i.e., pixel level or the connected component (CC) level) and clusters the neighboring elements to form larger homogeneous clusters until no more homogeneous zones can be built.

Examples of top-down approaches, also called knowledge-based approaches, are the XY cut algorithm [10] and white-space algorithm [11]. The algorithms depend on computing the horizontal and vertical histograms of the foreground pixels and finding the maximum number of large white rectangles. The top-down approach may be the most time-efficient approach in cases of noise-free and skew-free documents with "Manhattan layouts" (rectangular zones) and plain backgrounds.

Examples of bottom-up approaches, also called data-driven approaches, are the Docstrum algorithm [12], Voronoi algorithm [13], ridge-based constrained text line detection [14], the run-length smearing algorithm (RLSA) [15], and others [16,17]. These approaches work well in cases where the top-down approach fails like arbitrary/complex shape layouts, skewed or curved document images, and noisy or degraded images. The algorithms may have a considerably long processing time when the number of clustered elements on the page is large.

Researchers have proposed modifications to overcome the drawbacks of basic top-down and bottom-up approaches (e.g., [14,18]). Two additional variants for separating text and non-text page zones have been proposed: (1) A hybrid approach that has both top-down and bottom-up elements [19], and (2) a multi-scale resolution algorithm where text feature maps are analyzed for the same document image at different resolutions [20].

2.2 Layout segments classification

For the second step of physical layout analysis, the task of classifying the derived document zones into certain categories, various solutions exist. Some researchers proposed the use of *k*-means clustering and heuristic rules for the classification task [21]. Others prefer machine learning tools, such as neural networks and support vector machines, to distinguish text and graphic zones [3,5,22], or different types of text zones, such as main text and handwritten side notes in historical documents [4] or handwriting in text boxes of official printed forms [23].

Some researchers reversed the traditional order of the two steps of physical layout analysis [22,24,25]: Their methods first apply machine learning tools to the pixels or connected components of the original document and classify them as representing text or non-text, and then build up homogeneous clusters or zones. The approach we propose in this paper also uses a reversed order for PLA. Our method first classifies small image regions as containing text or non-text and then groups neighboring patches of the same class into larger image zones.

Our approach to use an ensemble of classifiers was motivated by an observation by Rahman and Fairhurst [26] that combining multiple expert (classifier) decisions can produce more robust, reliable, and efficient recognition performance than using a single expert classifier. They also warned that a single classifier with a single feature set and a single generalized classification strategy often does not comprehensively capture the large degree of variability and complexity encountered in many practical task domains. Accordingly, we experimented with an extensive feature set to feed into our classifier ensemble.

2.3 Layout analysis datasets

In the literature, the datasets used for DLA system construction and evaluation vary widely. The number of document images in a dataset is typically in the order of hundreds. Some researchers used publicly available datasets like UW-III [27], UNLV [28], IUPR [20], DFKI-1 [29], or subsets of these that are appropriate to the task at hand. Other researchers collected their own private datasets, which are typically limited in size or variability (e.g., [3,30]). For the task we addressed in this paper, localizing and classifying text and graphic regions in Arabic document images, BCE-Arabic-v1 [6] is the only dataset we know that is publicly available.

2.4 Layout analysis of Arabic documents

The task we address in this paper, layout analysis of scanned pages of printed modern books and newspapers, has not received much attention in the literature. Research on layout analysis of Arabic documents has mostly focused on historical, handwritten manuscripts [4,22,31,32], which pose different challenges (document degradation, ink bleeds, elaborate decorations, etc.). We here highlight the work by Boussellaa et al. [32], who performed multi-scale image analysis to segment image blocks and then classify them as belonging to the foreground or background. Misclassified blocks are corrected and smoothed using 8-nearest-neighbors pixel voting. A fuzzy *c*-means classifier discriminates text pixels from graphics pixels for the obtained foreground pixels. Our methodology differs in that we do not differentiate foreground and background specifically and is similar in the use of image blocks and 8-nearest-neighbor voting. We also note the work by Belaïd and Ouwayed [31] who analyze small subimages with Fourier descriptors to determine hand writing orientation in text-only manuscripts. Our proposed system also works on small subimages ("patches") and uses a Fourier descriptor among others.

Few efforts described in the literature are directed to documents with printed Arabic script and complex layouts [3,5,33]. Bukhari et al. [33] presented a document layout analysis system for Arabic documents that performs page decomposition (using multi-resolution morphology-based segmentation), text line detection (using anisotropic Gabor filter image smoothing and ridge detection with white-space finding), and detection of the reading order. They modified a multi-resolution method by Bloomberg [34] and used a white-space-finding algorithm by Breuel [35]. What is remarkable about their work is the robustness of their textline detection method regardless of the amount of border noise present in the image. The proposed algorithm was shown to outperform the *X*–*Y* algorithm on a dataset containing 25 Arabic and 20 Urdu documents.

Hadjar and Ingold focused on Arabic newspapers with complex layouts. Their initial work [36] presented methods for thread recognition, frame recognition, image text separation, text line recognition, and line merging into blocks. In subsequent work [5], they created the semiautomatic platform PLANET using neural networks. They applied their previous page decomposition technique and then followed it with an interactive correction phase, where users are able to interactively correct over and under-segmentation errors generated in the previous phase. Their dataset includes 50 pages from three different newspapers (Annahar, AL Hayat, and AL Quds).

To achieve Arabic newspaper layout analysis, Alshameri et al. [3] used run-length smoothing for block building and a support vector machine for labeling connected components as text or non-text. The logical reading order of text lines was defined using modified topological sorting rules. The combined method contains various heuristic rules and was tested on 50 images.

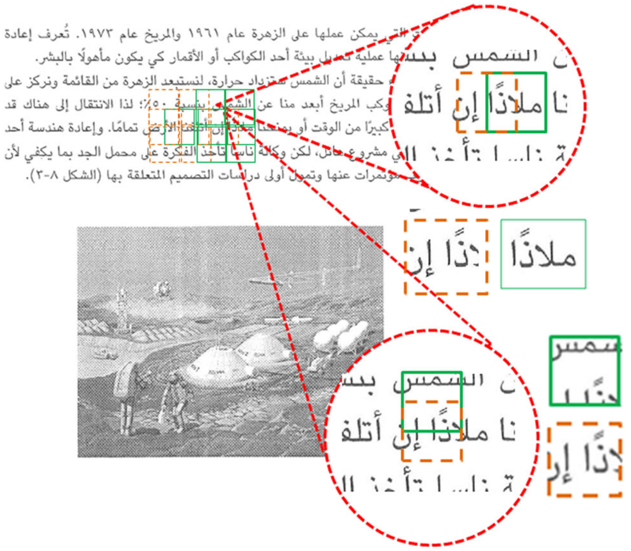
The three systems for Arabic newspaper layout analysis described above predate the large publicly available BCE-Arabic-v1 [6] dataset. They were tested on relatively small proprietary datasets (45 or 50 pages), for which they achieved high classification results (97–99%). Note that the subset of the BCE-Arabic dataset that we use here is one order of magnitude larger, which helps with overfitting issues.

2.5 State of the art in layout analysis

Recent publications on physical layout analysis show that progress is made with three types of approaches: deep learning, classical machine learning, and traditional rule-based image processing. None of these approaches can solve the general problem of layout analysis for all document domains.

Deep networks have been proposed to solve specific layout analysis problems like text line extraction in historical documents [37–39], table extraction [40], newspaper article segmentation [41], layout analysis of scientific publications [42], census record counting [43], and logical analysis of born-digital documents [44]. Deep networks that interpret layout shape rather than components have been proposed to categorize documents [45–48], e.g., letters, resumes, invoices. Deep learning has also been used to extract document features that are then interpreted for layout classification tasks by traditional machine learning approaches that employ SVM classifiers [49–51].

The state of the art that uses traditional image processing techniques includes rule-based morphological methods that segment Chinese documents with complex layouts using top-down image projections [52], detect textlines in degraded English historical documents [53], and extract text in historical Tibetan document images [54]. A recent publication that uses a combination of morphological tools and traditional machine learning is the work by Hesham et al. [8]. Their method can extract textlines in modern and historical Arabic books using an SVM to classify text and non-text zones. For such a two-class problem, an SVM classifier is appropriate because it maximizes the margin around the separating hyperplanes. Among the many related works that we discussed here, only Hesham et al. consider the same problem as we do. Instead of using one SVM, however, we use an ensemble. Another difference is that our method does not first segment zones and then classify them but instead first classifies small, overlapping patches that are then gathered to build homogeneous zones.



ت التي يمكن عملها على الزهرة عام ١٩٦١ والمريخ عام ١٩٧٣. تُعرف إعادة
ها عملية تعديل بيئة أحد الكواكب أو الأقمار كي يكون مأهولًا بالبشر.
حقيقة أن الشمس ستزداد حرارة، لنستبعد الزهرة من القائمة ونركز على
وكب المريخ أبعد منا عن الشمس بنسبة ٥٠٪؛ لذا الانتقال إلى هناك قد
كبيرًا من الوقت أو يمنحنا ملاذًا إن أتلفنا الأرض تمامًا. وإعادة هندسة أحد
لي مشروع هائل، لكن وكالة ناسا تأخذ الفكرة على محمل الجد بما يكفي لأن
مؤتمرات عنها وتمول أولى دراسات التصميم المتعلقة بها (الشكل ٨-٣).

Fig. 1 Creating patches in a document from BCE-Arabic-v1 that contains text and images. The enlarged details show neighboring patches with 50% vertical and horizontal overlap. In each case, a patch with a dashed (orange) outline overlaps another patch with a solid (green) outline (color figure online)

3 Materials and methods

Our method analyzes the physical (geometric) layout of a grayscale document image by working with small square-shaped subimages, called patches. Since we use an ensemble of classifiers to process the document patches, we call our solution ECDP—ensemble-based classification of document patches. Examples of overlapping patches in the text zone of an Arabic book page are shown in Fig. 1. In this section, we first describe the document datasets and then explain our method.

3.1 Datasets

We report experiments with three datasets of raster-image Arabic documents. The largest, BCE-Arabic [7], is a collection of 1833 annotated documents. We used a 567-document subset of this collection, which is defined by all the documents in the database that contain both text and images. In particular, we used 383 document pages to train and test the system presented in this paper, and the subset of data that contain both text and graphic elements (charts or diagrams), 184 pages, to evaluate the robustness of our system to a new type of page content. The remaining documents in BCE-Arabic are text-only documents. We excluded them to ensure a balance between the number of text and graphics training patches and prevent our machine learning system to simply learn anomalies.

The second dataset is a proprietary collection by Hesham et al. [8]. It contains 85 Arabic document pages and comprises

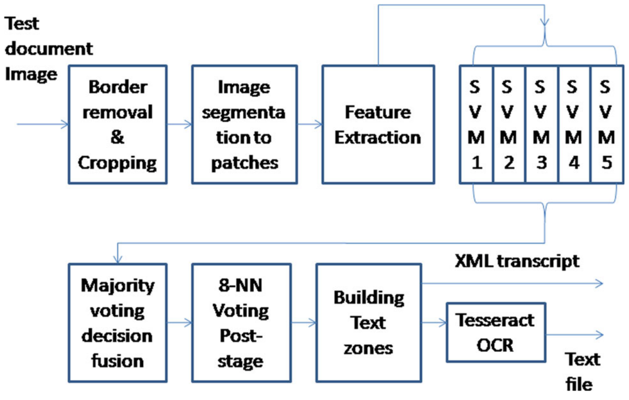


Fig. 2 Block diagram of the proposed ECDP system

of 60 book pages, 10 scanned historical documents, and 15 magazine pages. The third dataset, provided by RDI [9], is a proprietary collection that contains 109 Arabic book and magazine pages. We used the Hesham and RDI datasets only for testing, not for training of our system.

Ground truth for the location and type of layout zones in the documents of BCE-Arabic [7] were provided in XML files by Saad et al. [6]. The ground-truth segmentations, used by Hesham et al. [8] to evaluate their SVM system on their dataset as well as the RDI dataset, were not made available to us. We therefore needed to recreate ground-truth annotations for the Hesham and RDI datasets. We used the Alethia tool [55] to segment the document image into text paragraphs and graphics zones. We even segmented marginal text like headers, footers, and page numbers.

3.2 ECDP system overview

The ECDP system pipeline has seven steps (see Fig. 2: (1) Image borders due to the document scanning process are removed and a cropped image of the document is obtained. (2) Overlapping patches are created in the cropped image and their features extracted. (3) Patches are classified as "text" or "non-text," based on the interpretation of their features by the majority of an ensemble of SVM classifiers. (4) Patches may be re-classified if their class differs from the class of the majority of the surrounding patches, which are the eight immediate neighbors. The majority rule is also used to resolve potentially occurring conflicts when overlapping patches have opposite class labels. (5) Patches of the same class are combined into larger image zones. (6) The resulting text zones are passed to the Tesseract OCR package (https://github.com/tesseract-ocr) to obtain a machine-accessible transcript, stored in the first ECDP output file. (7) A second output file is produced in XML format that contains the location information of the zones and thus links to the searchable text stream for each text zone.

3.3 Training methodology and data labels

To choose an appropriate model for our ECDP system from numerous rivaling trained models, we conducted extensive validation experiments on these models. We built our system using 383 documents with text and graphic elements. We divided them into 250 pages for training, 33 pages for validation, and 100 pages for testing. Since our classifiers use the features extracted from image patches (and not full documents) and we use overlapping patches, the actual numbers of data points used for training, validation, and testing depend on the patch size and the patch overlap ratio. The smaller the patch size is chosen, the larger the training and validation sets become. With an overlap ratio of 50%, they range from tens of thousands to hundreds of thousands of patches (see Table 1). The ground-truth label for each patch is automatically retrieved from the ground-truth labels of the zone that the patch belongs to.

Our idea to use of overlapping patches is noteworthy because it serves as a data augmentation step. A 50% overlap, specifically, ensures sufficient augmentation without providing too much redundancy.

Section 4 describes the experiments we conducted to investigate the best patch size to use in our system. Section 4 also describes our experiments on the validation dataset to discover which type of patch features to select.

The data files are available in the form of gray scale jpg images and corresponding XML annotations in PAGE format [56]. The annotations include information of the foreground regions, such as the bounding box of the region and its label, i.e., a "text" or "image" region.

3.4 Features

To find out the most descriptive feature set to be used for ECDP with image patch input, we investigated four types of features:

- *Edge operators:* Canny (Fig. 3), Sobel, Prewitt, Roberts, and the Laplacian of Gaussian approximation;
- *Transforms*: Fourier (FFT), discrete cosine (DCT), Hough, and Radon;
- *Texture features*: Gray-level range and standard deviation, entropy of the gray-level distribution, and the properties of the gray-level co-occurrence matrix (GLCM): contrast, correlation, energy, and homogeneity;
- *Moments*: Hu's seven moments and Zernike moments.

These features are particularly suited for patch analysis, in particular, distinguishing patches that contain text fragments versus graphics regions.

Canny's edge detection algorithm is known to find both strong and weak edges, the Laplacian gradient enhances both

Table 1 Number of training and validation examples for different patch sizes (empty background patches are excluded)

Size	30 × 30	50 × 50	60 × 60	70 × 70	80 × 80	90 × 90
Training	883,031	289,214	172,623	113,344	87,339	74,133
Validation	212,530	83,302	59,482	44,725	34,940	28,438

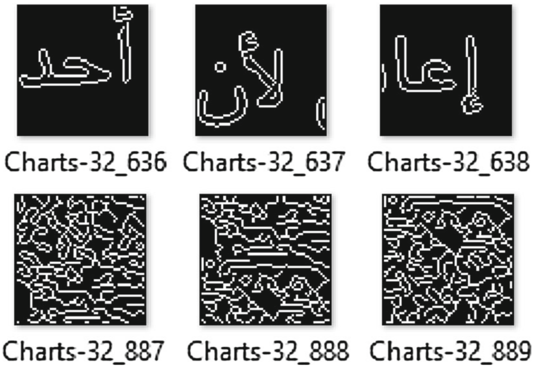


Fig. 3 Canny edge feature for patches of text (top) and graphic (bottom)

the low and high contrast information (e.g., [57]). The Fourier and DCT coefficients may serve as shape descriptors to the patch content [58]. The Hough transform can be used to detect the lines in the image patch, and the Radon transform to compute projections onto different rotation axes. The GLCM has been used successfully to measure texture [59,60].

Even the smallest patches we tested contain a large number of pixels that can be used for measuring a feature. Due to the grid-based sampling of the image into patches, a patch may contain an incomplete connected component (e.g., a part of a character) or multiple connected components (characters or parts of characters). A patch may contain arbitrarily sized, rotated, or translated connected components.

A patch that does not contain any connected components is deemed to belong to the document background. Features are only computed for foreground patches.

3.5 Training stage

The training stage starts by parsing the annotation XML file and extracting the bounding box information of each foreground region. Each foreground region is segmented into adjacent overlapping patches according to the pre-defined patch size. Features vectors are extracted for each patch, labeled as text or non-text, and used as training examples for the selected classifier.

We designed a five-SVM-classifier ensemble by partitioning the training documents into five cohorts of 50 documents. We needed an odd number of classifiers so that we can use majority voting to fuse the decisions of the ensemble. Five classifiers can capture the data variance better than three and are computationally cheaper to use than seven. Each SVM classifier was trained using patches from the 50 documents in its corresponding cohort. We conducted experiments (see Sect. 4) on the validation data to tune SVM parameter values and select the best-performing SVM kernel (linear, polynomial, RBF).

3.6 Test stage

Each document image in the test dataset is processed by our ECDP system as shown in Fig. 2. An example image and a visual representation of the ground truth labels of the image patches, which we created by parsing the annotations of the input image as provided by BCE-Arabic-v1 [6], are shown in Fig. 4a, b, respectively. The results by the five SVMs of ECDP are shown in Fig. 4c–g. A second example is shown in Fig. 5, which shows the regions of a document image that were classified to contain text, the OCR transcription of this text, and the corresponding XML output file.

3.7 Evaluation metrics

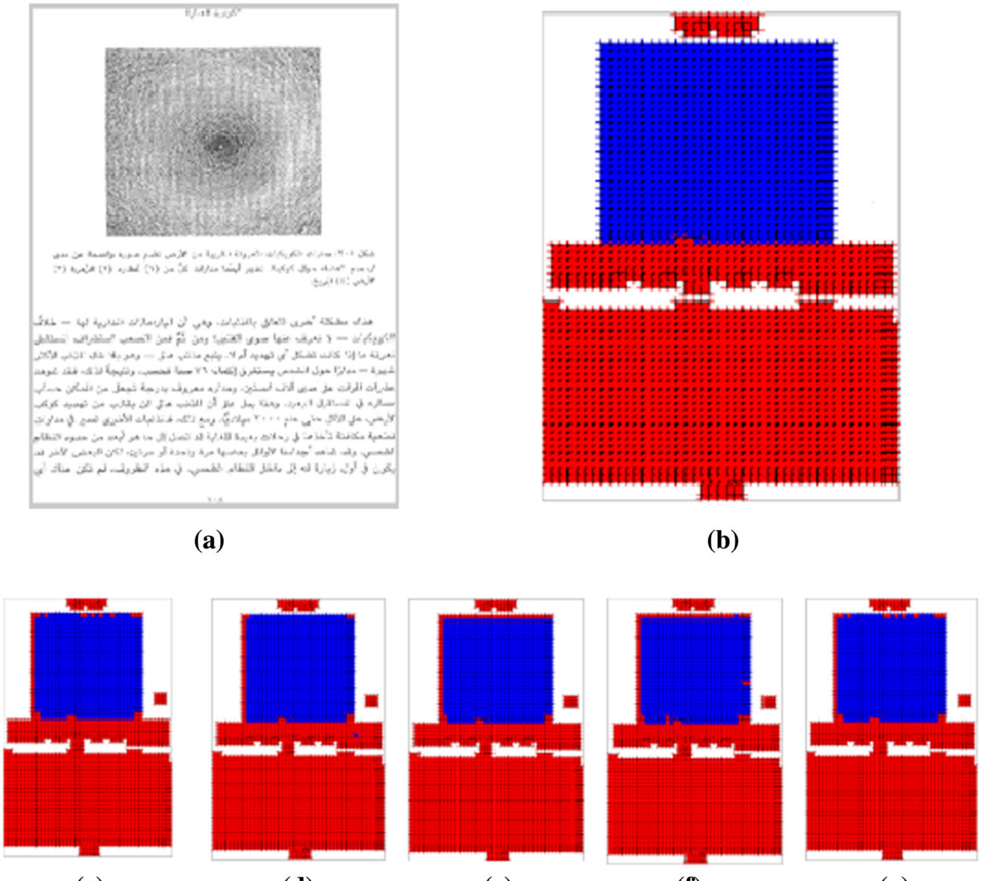
We used two performance evaluation metrics to judge the classification performance of ECDP. The first is the binary classification accuracy A of foreground document patches:

$$A = \frac{C_{\mathrm{g}} + C_{\mathrm{t}}}{T_{\mathrm{f}}}, \tag{1}$$

where C_{g} is the number of correctly classified graphic patches, C_{t} the number of correctly classified text patches, and T_{f} the number of foreground patches.

The number T_{f} of detected foreground patches might exceed the number of annotated patches if residual noise (e.g., an ink stain) escaped the noise removal stage of ECDP. An example is seen in Fig. 4, where the SVMs classified a noise patch to the right of the image on the page as text (red). To deal with this problem, our second evaluation metric measures accuracy only with respect to the class of interest, which are text patches. We use the F_1-score, which accounts for precision P and recall R:

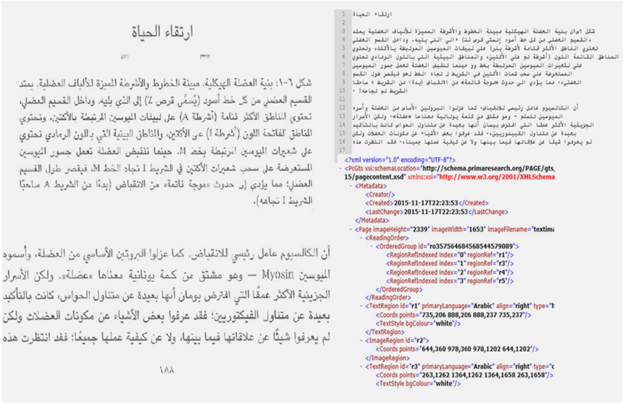
$$F_1 = \frac{2 \cdot PR}{P + R}, \quad \text{and} \quad P = \frac{C_{\mathrm{t}}}{(C_{\mathrm{t}} + F_{\mathrm{t}})}, \quad R = \frac{C_{\mathrm{t}}}{(C_{\mathrm{t}} + M_{\mathrm{t}})}, \tag{2}$$



(a) (b)

(c) (d) (e) (f) (g)

Fig. 4 Visual representation of the reference patch labels and the 5-SVM output labels of a test document image. Red image regions represent patches classified as "text" (lighter gray in non-color media), blue as "graphic" (darker gray) (color figure online)



ارتقاء الحياة

شكل ٦-١: بنية العضلة الهيكلية، مبينة الخطوط والأشرطة المميزة لألياف العضلية. يمتد
القسيم العضلي من كل خط أسود (يُسمَّى قرص Z) إلى الذي يليه. وداخل القسيم العضلي،
تحتوي المناطق الأكثر قتامة (أشرطة A) على لييفات الميوسين المرتبطة بالأكتين، وتحتوي
المناطق الفاتحة اللون (أشرطة I) على الأكتين، والمناطق البينية التي بـاللون الرمادي تحتوي
على شعيرات الميوسين المرتبطة بخط M. حينما تنقبض العضلة تعمل جسور الميوسين
المستعرضة على سحب شعيرات الأكتين في الشريط I تجاه الخط M، فيقصر طول القسيم
العضلي؛ مما يؤدي إلى حدوث «موجة قاتمة» من الانقباض (بدءًا من الشريط A ساحبًا
الشريط I تجاهه).

أن الكالسيوم عامل رئيسي للانقباض. كما عزلوا البروتين الأساسي من العضلة، وأسموه
الميوسين Myosin — وهو مشتق من كمة يونانية معناها «عضلة». ولكن الأسرار
الجزيئية الأكثر عمقًا التي افترض بومان أنها بعيدة عن متناول الحواس، كانت بالتأكيد
بعيدة عن متناول الفيكتوريين؛ فقد عرفوا بعض الأشياء عن مكونات العضلات ولكن
لم يعرفوا شيئًا عن علاقاتها فيما بينها، ولا عن كيفية عملها جميعًا؛ فقد انتظرت هذه

١٨٨

```
ارتقاء الحياة

شكل ٦-١: بنية العضلة الهيكلية مبينة الخطوط والأشرطة المميزة للألياف العضلية يمتد
،القسيم العضلي من كل خط أسود إيسمى قرص ئذ) •الى التي يليه. وداخل القسم العضلي
تعتري الناطق الأكثر قتامة أشرطة بنر) على لييفات الميوسين المرتبطة بالأكتد، وتحتوي
المناطق الفاتحة اللون (أشرطة لم على الأكتين، والمناطق البينية التي باللون الرمادي تحتوي
على لشعيرات الميوسين المرتبطة بخط وو حينما تنقبض العضلة تعمل جسور الميوسين
المستعرضة على سحب شعات الأكتين في الشريط ل تجاه الخط لهم فيقصر طول القسم
العضلي،• مما يؤدي الى حدوث «موجة قاتمة» من الانقباض (بدءا من الشريط ٨ ساحبا
الشريط لم تجاهه( •

أن الكالسيوم عامل رئيسي للانقباض• كما عزلوا البروتين الأساس من العضلة وأسره
الميوسين لتسلم - وهو مشتق من كلمة يونانية معناها «عضلة»• ولكن الأسرار
الجزيئية الأكثر عمقا التي افترض بيمان أنها بعيدة عن متناول الحواس كانت بالتاكيد
بعيدة عن متناول الفيبتوريين،• فقد عرفوا بعض الأشياء عن مكونات العضلات ولكن
لم يعرفوا شيئا عن علاقاتها فيما بينها ولا عن كيفية عملها جميعا،• فقد انتظرت هذه
```

```
  <?xml version="1.0" encoding="UTF-8"?>
- <PcGts xsi:schemaLocation="http://schema.primaresearch.org/PAGE/gts,
  15/pagecontent.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema
  - <Metadata>
      <Creator/>
      <Created>2015-11-17T22:23:53</Created>
      <LastChange>2015-11-17T22:23:53</LastChange>
    </Metadata>
  - <Page imageHeight="2339" imageWidth="1653" imageFilename="textima
    - <ReadingOrder>
      - <OrderedGroup id="ro357564684568544579089">
          <RegionRefIndexed index="0" regionRef="r1"/>
          <RegionRefIndexed index="1" regionRef="r3"/>
          <RegionRefIndexed index="2" regionRef="r4"/>
          <RegionRefIndexed index="3" regionRef="r5"/>
        </OrderedGroup>
      </ReadingOrder>
    - <TextRegion id="r1" primaryLanguage="Arabic" align="right" type="h
        <Coords points="735,206 888,206 888,237 735,237"/>
        <TextStyle bgColour="white"/>
      </TextRegion>
    - <ImageRegion id="r2">
        <Coords points="644,360 978,360 978,1202 644,1202"/>
      </ImageRegion>
    - <TextRegion id="r3" primaryLanguage="Arabic" align="right" type="c
        <Coords points="263,1262 1364,1262 1364,1658 263,1658"/>
        <TextStyle bgColour="white"/>
```

Fig. 5 Sample output of our ECDP system: extracted text zones (left), Tesseract OCR output (upper right), and XML file (lower right)

where F_t is the number of patches misclassified as text (false alarms) and M_t the number of text patches misclassified as graphic patches (false negatives).

4 Experiments and results

This section describes two sets of experiments with ECDP and one comparison experiment with the run-length smearing algorithm (RLSA) [15]. The first set of experiments was conducted to select features and tune ECDP system parameters using the validation dataset (Sect. 4.1). The resulting best-performing model was then used to conduct ECDP evaluation experiments on three test datasets (Sect. 4.4), which show system robustness toward new document layouts and contents.

4.1 Methodology of validation experiments

We first tested each document in the validation dataset individually. For all documents in the validation dataset, we then computed four statistics of the patch classification accuracy A: average, standard deviation, maximum, and minimum value.

To find the best-performing set of features among the options we considered, we fixed other experimental param-

Table 2 Patch classification accuracy (%) statistics for edge-descriptor features

Fusion	Canny		Sobel		Roberts		Laplacian		Prewitt	
	Avg.	Vote	Avg.	Vote	Avg.	Vote	Avg.	Vote	Avg.	Vote
Avg.	87.10	**92.79**	57.63	69.26	74.31	83.30	82.51	88.69	57.90	69.40
SD	7.68	5.68	13.96	11.54	8.16	4.46	7.62	4.91	13.90	11.53
Max.	95.24	98.57	89.80	92.14	89.56	91.94	92.70	95.45	89.69	92.00
Min.	63.79	79.71	2.90	21.61	41.44	65.42	54.93	68.59	2.87	21.61

Table 3 Patch classification accuracy (%) statistics for transform features

Fusion	FFT		DCT		Hough		Radon	
	Avg.	Vote	Avg.	Vote	Avg.	Vote	Avg.	Vote
Avg.	87.78	**92.43**	60.85	72.70	82.43	84.88	82.16	89.53
SD	8.54	5.18	11.64	8.08	9.52	5.55	12.06	9.09
Max.	97.04	99.03	92.18	94.34	94.47	95.29	91.98	97.32
Min.	62.08	77.52	22.83	53.81	50.38	70.29	43.64	54.86

eters, such as patch size (90 × 90) and the SVM kernel type (polynomial) with the default parameter values of the LIBSVM library (no further fine tuning was performed to avoid overfitting to the current dataset). We compared the accuracy statistics using the features introduced in Sect. 3.4: five edge, four transform, and seven texture descriptors, as well as two kinds of moment descriptors. We also tested a combination of texture features (gray-level range and standard deviation, entropy) and a combination of the four GLCM statistics. We evaluated the low-order and high-order Zernike moments separately and in combination. Furthermore, we paired the best-performing feature of each of the four feature types, which yields six pairs, grouped the three best-performing features of each type, which yields three triplets, and finally combined the best-performing features of all four types into a single quadruple. As a result, we tested 10 combined sets of features.

We experimented with two fusion schemes to combine the decisions of the ensemble of five SVM classifiers: (1) averaging and (2) majority voting. It was quickly obvious when testing edge and transform features that majority voting outperformed averaging, so the remaining validation experiments used majority voting only.

We then kept features and patch size fixed and compared the performance of three kernel types. We also tested the best two kernels with seven sizes of patches.

We tested the impact of the re-classification, step (4) of the ECDP system pipeline, which re-classifies a patch if the majority of surrounding patches belong to a different class. In this ablation experiment, we compared the statistics of A and F_1 when ECDP is used with and without this step.

4.2 Implementation details

Extracting edge features or Fourier/DCT transform coefficients from a patch yields 2D feature matrices of the same size as the patch. Reshaping an $n \times n$ matrix to a 1-dimensional $1 \times n^2$ input vector to train the SVM classifiers is computationally expensive. A computationally more efficient process is to concatenate the vertical and horizontal summation of the $n \times n$ feature matrix to produce a $1 \times 2n$ input vector. This modification generated size and translation-invariant feature representations and gave us similar performance.

The Hough transform matrix is typically very sparse. A 90 × 90 text patch, resulting in a 505 × 360 = 181,800 element matrix, for example, had 146,928 zeros. For efficiency, our method therefore concatenates the vertical and horizontal summations of the Hough matrix of a patch, producing two 1 × 360 length feature vectors.

The Radon transform produces 1D output, and validation experiments were done using the projections of a patch on axes at 0, 45, 90, 135, 180, 225, 270, and 315 degrees. The texture filters (gray-level range and standard deviation, entropy) applied to a $n \times n$ patch yield $1 \times n$ feature vectors. Each GLCM statistics (contrast, correlation, energy, homogeneity) is just one scalar. The magnitudes of Zernike moments were extracted for low (32 elements) and high-order (32 elements) values with repetitions, as explained by Tahmasbi et al. [61].

4.3 Results of validation experiments

The results of our extensive validation experiments with single features and their combinations, as described in Sect. 4.1, are given in Tables 2, 3, 4, 5 and 6 (notable numbers in boldface). Canny edge features yielded by far the most accurate average performance among five edge descriptors (Table 2) and the Fourier transform among four transforms (Table 3).

Table 4 Patch classification accuracy (%) statistics for texture features

	Texture filters				GLCM statistics				
	Range	SD	Entropy	All	Contr.	Corr.	Energy	Homog.	All
Avg.	89.18	88.08	92.00	92.77	**93.19**	92.31	91.30	91.29	92.53
SD	8.69	6.14	7.63	8.27	7.58	8.97	8.60	8.60	7.63
Max.	96.51	95.37	98.86	99.76	99.76	99.43	99.43	99.43	99.59
Min.	55.50	66.50	69.00	66.73	72.28	68.00	67.09	68.00	72.55

Table 5 Patch classification accuracy (%) statistics for moments features

Moments	Hu	Zernike		
	All 7	Low order (3:10)	High order (10:17)	All 64 moments
Avg.	**88.18**	82.16	79.62	80.65
SD	10.56	5.34	6.78	6.52
Max.	97.40	89.20	89.61	89.77
Min.	47.70	66.88	63.80	64.45

Table 6 Patch classification Accuracy (%) statistics for different feature set combinations: Set 1: Canny and FFT; Set 2: Canny, FFT and 3 texture filters; Set 3: Canny, FFT and GLCM contrast; Set 4: Canny, FFT, 3 texture filters and GLCM contrast; Set 5: FFT, 3 texture filters and GLCM contrast; Set 6: FFT and 3 texture filters; Set 7: FFT and GLCM contrast; Set 8: Canny, 3 texture filters and GLCM contrast; Set 9: Canny and 3 texture filters; Set 10: Canny and GLCM contrast

Set	1	2	3	4	5	6	7	8	9	10
Avg.	**94.25**	93.65	94.23	93.64	92.91	92.85	93.59	93.14	92.98	93.48
SD	6.11	6.84	6.15	6.83	7.39	7.45	6.27	7.07	7.13	5.81
Max.	99.76	99.51	99.76	99.51	99.27	99.27	99.51	99.35	99.27	99.19
Min.	80.18	75.06	79.92	75.06	72.91	72.51	74.94	72.73	72.82	80.36

The nine texture features produced similar results, with the GLCM contrast slightly winning (Table 4). A combination of all seven Hu moments yielded more accurate results than each one individually or combinations of Zernike moments (Table 5). The Hu feature combination, however, did not outperform the best edge and transform features, and so moment features were not further tested.

We presume that the Fourier transform and Canny edge features outperform the other features in distinguishing patches with Arabic script from patches without because they can represent high-frequency information and long edges without gaps. We also note that the performance of the features is highly correlated (between 0.72 and 0.89). In particular, the correlation of accuracy values between the Canny and FFT descriptors was 0.89. We measured this by comparing the accuracy values obtained per document of the validation dataset for two features (Fig. 6).

Experiments of the 10 feature combinations (Table 6) revealed that a combination of Canny edge features and Fourier transform magnitude and frequency descriptors yielded the most accurate results on our validation dataset. This means that, while there is a correlation between the performance when the Canny and FFT descriptors are applied separately, applying them in combination still left sufficient room for performance improvement. For example, we discovered that, for documents with text and line drawings or diagrams with dotted backgrounds (halftones) and gradient shading, the Canny descriptor yielded several percent points higher-accuracy results than the FFT descriptor. For such documents, the combination of Canny and FFT descriptors yielded even higher-accuracy results.

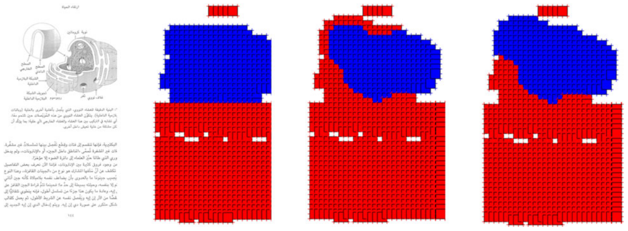


Fig. 6 Example of correlated performance results for Canny and FFT descriptors: (1) original, (2) ground truth with text (red or light gray) and graphic (blue or dark gray) patches, and results based on (3) Canny descriptor only (accuracy 87.7%), and (4) FFT only (accuracy 88.5%) (color figure online)

The RBF SVM kernel yielded a slightly higher average accuracy than the polynomial or linear kernels for the patch size 90 × 90 (Table 7). The results for polynomial and RBF kernels are similar for other patch sizes and generally degrade

Table 7 Patch classification accuracy (%) statistics for different SVM kernels

	90 × 90		
	Poly	RBF	Linear
Avg.	94.25	**94.31**	93.52
SD	6.11	6.16	6.75
Max.	99.76	99.84	99.59
Min.	80.18	80.18	77.24

Table 9 Effect of the re-classification step on $F1$-score and patch classification accuracy statistics

	Re-classification			
	Without		With	
	$F1$-score	Accuracy	$F1$-score	Accuracy
Avg.	91.03	95.18	93.56	96.91
SD	14.48	5.14	12.89	4.67
Max.	99.70	99.66	100.00	100.00
Min.	40.00	83.22	49.10	80.26

for large patches, 100×100 (Table 8). The 70×70 RBF kernel model was selected for the ECDP system (very comparable to the highest performing model '60 × 60 Poly') because it enabled faster training and tuning of fewer parameters.

The ablation experiment, which evaluates the effect of the patch re-classification step (step 4 of ECDP), shows that this step improves the average F_1-score and accuracy A only by a percent point or two (Table 9). The power of the re-classification step lies in its benefit of creating homogeneous text and non-text regions for the subsequent zone-building step, as shown in Fig. 7.

The final model used for our ECDP system processes Canny and FFT features of 70 × 70 size patches and uses the RBF SVM kernel, majority voting for the ensemble, and patch re-classification. With this model, 28 of the 33 validation documents (84.8%) have a patch classification accuracy of $\geq$ 95%, and only three $\leq$ 87% (Fig. 8).

Visual inspection of the ECDP-processed validation data showed that some border noise and ink stains remained. Visual inspection also revealed that cases with low performance had particularly challenging non-text regions with line drawings, faded illustrations, or rotated graphics.

Low performance was also measured due to insufficiently fine-grained labeling of the ground truth. We found that annotators often chose not to label small text regions within maps and diagrams as text. For example, the ground-truth annotation of the bottom image shown in Fig. 9 was too coarse. The rotated text in the legend of the linedrawn map was not hand-labeled as text but detected by ECDP as text.

Although trained on Arabic documents, ECDP was able to detect some English text (Fig. 10).

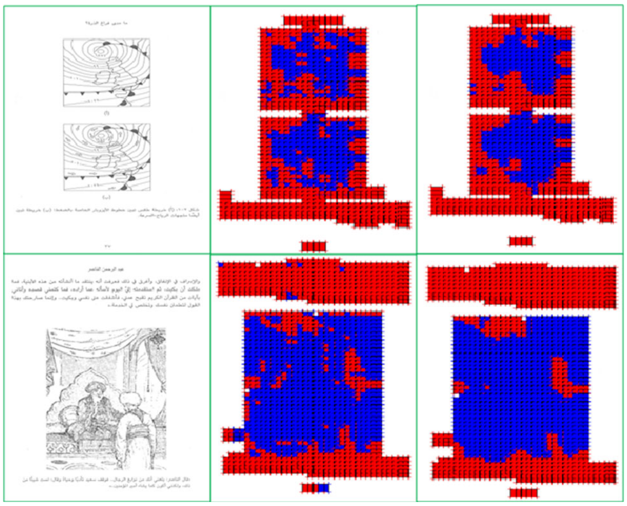


Fig. 7 Two examples of the effect of the re-classification step: original image (left), SVM fusion output (center), and re-classification output (right)

4.4 Methodology and results of experiments on test data

We used our final model of ECDP (patches with 70 × 70 pixels and 50% overlap, RBF SVM kernel, Canny and FFT features) for our experiments on test data. Our MATLAB implementation of ECDP processes each document in 1:14 min on average. This run time can be reduced by recoding the system in a high-level language and code optimization.

Table 8 Patch classification accuracy (%) statistics for different patch sizes

	100 × 100		80 × 80		70 × 70		60 × 60		50 × 50		40 × 40		30 × 30	
	Poly	Rbf	Poly	Rbf	Poly	Rbf	Poly	Rbf	Poly	Rbf	Poly	Rbf	Poly	Rbf
Avg.	67.19	67.19	94.05	94.12	95.25	**95.18**	**95.30**	95.25	95.14	94.76	94.90	93.79	94.36	93.76
SD	14.43	14.43	5.52	5.54	5.13	5.14	4.94	4.72	5.12	4.20	4.63	3.30	4.60	3.67
Max.	91.02	91.02	99.42	99.35	99.85	99.66	99.75	99.47	99.56	98.68	98.85	97.04	98.45	97.49
Min.	27.50	27.50	81.87	81.14	83.43	83.22	84.04	83.39	81.15	83.13	83.09	85.27	83.86	84.98

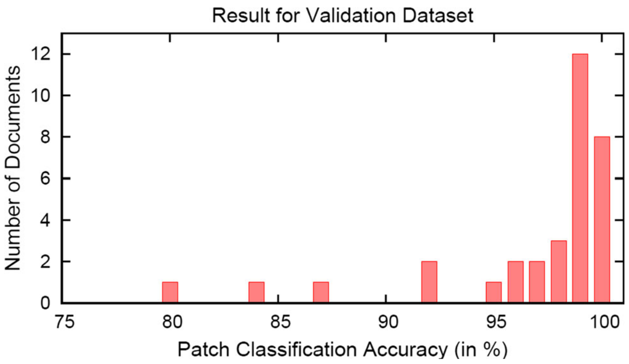


Fig. 8 Histogram of accuracy results for validation dataset documents

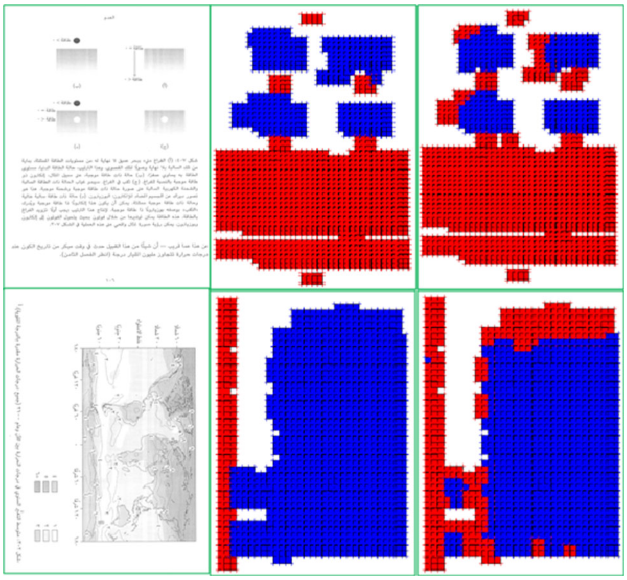


Fig. 9 Two challenging validation documents with line drawings and faded-rotated graphic: original image (left), ground truth from BCE-Arabic-v1 (center), and ECDP system output (right). In the second example, the ECDP system was able to separate the rotated text embedded within the figure and so arguably provides a more valuable annotation than the ground-truth annotation

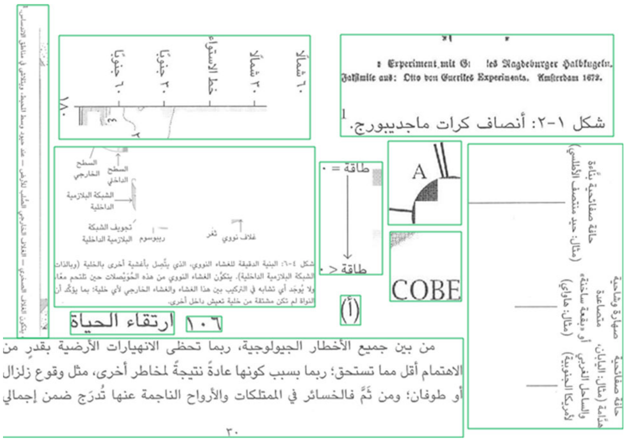

٦٠ شمالاً
٣٠ شمالاً
خط الاستواء
٣٠ جنوبًا
٦٠ جنوبًا
١٨٠
Experiment mit G: les Magdeburger Halbkugeln.
Facsimile aus: Otto von Guericke Experimenta. Amsterdam 1672.
شكل ١-٢: أنصاف كرات ماجديبورج.
السطح الخارجي
السطح الداخلي
الشبكة البلازمية الداخلية
تجويف الشبكة البلازمية الداخلية
ريبوسوم
ثغر
غلاف نووي
طاقة = ٠
طاقة < ٠
A
COBE
(أ)
ارتقاء الحياة
١٠٦
من بين جميع الأخطار الجيولوجية، ربما تحظى الانهيارات الأرضية بقدرٍ من
الاهتمام أقل مما تستحق؛ ربما بسبب كونها عادةً نتيجةً لمخاطر أخرى، مثل وقوع زلزال
أو طوفان؛ ومن ثَمَّ فالخسائر في الممتلكات والأرواح الناجمة عنها تُدرَج ضمن إجمالي
٣٠
حافة صفائحية بنّاءة (مثال: حيد منتصف الأطلسي)
حافة صفائحية متباعدة أو «بقعة ساخنة» (مثال: هاواي)
حافة صفائحية هدّامة (مثال: اليابان، والساحل الغربي لأمريكا الجنوبية)

Fig. 10 Examples of detected text, including English

Table 10 ECDP results on the test dataset

	*F*1-score	Accuracy
Avg.	95.26	97.3
SD	8.42	5.01
Max.	100.00	100.00
Min.	35.63	58.28

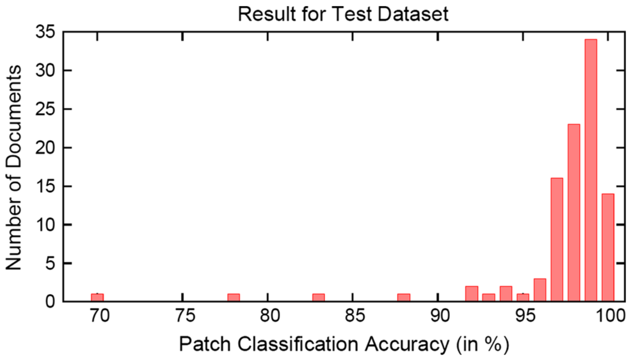


Fig. 11 Histogram of ECDP accuracy results for the 100 test documents from BCE-Arabic

We conducted six test experiments with ECDP, four with a different subset of the BCE-Arabic [7] and the remaining two with the Hesham [8] and RDI [9] datasets.

Our first experiment involves a comparison between the performance of ECDP and two other methods, the classical layout analysis method RLSA, short for run-length smearing algorithm [15], and the start-of-the-art commercial product (RDI-CleverPage) [62]. For this experiment, we used 100 of the 383 pages of BCE-Arabic-v1 that contain both text and images as our test set. We implemented the run-length smearing algorithm (RLSA) [15] with the following tuned thresholds: run-length for horizontal smearing tsh = 500 pixels, vertical smearing tsv = 800, horizontal smoothing = 30, and the thresholds for classifying text, ftr = 3, and fth = 3. We provided the company RDI our dataset and received the RDI-CleverPage was applied by RDI on our 100-page dataset, and computed zone location and type information was provided to us in XML files.

Note that ECDP was trained on 250 and validated on 33 *different* pages of this dataset. The 100 test documents contain 129,781 foreground patches (i.e., non-empty patches). ECDP classifies them with high average accuracy values (F_1 = 95.26% and A = 97.3), see Table 10. Ninety-one of the 100 test documents with text and images have a patch classification accuracy ≥ 95%, while only 4/100 have a patch classification accuracy ≤ 88% (Fig. 11).

Results of RLSA and RDI-CleverPage on our 100-page test dataset are summarized in Table 11 for segmentation performance and Table 12 for classification performance.

Our comparison of RLSA, RDI-CleverPage, and ECDP yielded the following results for the seven segmentation

Table 11 Segmentation performance of the classic RLSA, the state-of-the-art RDI- CleverPage (CP), and the proposed ECDP on a 100-page test dataset with text and graphic content, measured in terms of average black pixel rate (AvgBPR), over-segmentation error (OSE), under-segmentation error (USE), missed segmentation error (MSE), correct segmentation (CS), false alarm rate (FA), and block error rate (ρ)

	AvgBPR (%)	OSE	USE	CS	MSE	FA	ρ
RLSA	89.89	6.52	0.00	0.43	0.04	4.31	6.57
RDI-CP	91.37	0.67	0.20	0.43	0.15	0.57	1.02
ECDP	98.88	0.02	0.62	0.35	0.02	0.59	0.66

Table 12 Classification performance of RLSA, RDI-CleverPage, and ECDP with respect to the metrics precision (Pr), recall (Rec), $F1$-measure ($F1$) and average text versus non-text accuracy (Acc) over the test dataset of 100 document images with text and image content

	Pixels (%)							Blocks (%)						
	Text			Non-Text			Avg.	Text			Non-Text			Avg.
	Pr	Rec	$F1$	Pr	Rec	$F1$	PixAcc.	Pr	Rec	$F1$	Pr	Rec	$F1$	BlkAcc.
RLSA	55.07	99.35	70.86	99.90	88.81	94.03	90.09	96.17	81.03	87.95	25.58	66.92	37.02	79.78
RDI-CleverPage	32.7	96.43	48.84	99.34	72.99	84.15	75.80	72.30	71.47	71.88	52.52	53.55	53.03	64.82
ECDP	95.48	97.56	95.73	97.09	94.97	97.75	96.67	97.18	96.37	96.27	92.86	93.54	94.06	95.31

metrics, introduced by Shafait et al. [63] to compare segmentation algorithms:

- The average black pixel rate (AvgBPR) represents the number of black pixels contained in segmented blocks compared to the corresponding blocks in the ground-truth image. The rate is higher for ECDP than for RLSA and RDI-CleverPage, which indicates more consistently accurate segmentation.
- The over-segmentation error (OSE) compares the number of over-segmented blocks to the number of ground truth blocks. RLSA massively over-segmented our test data since it works on the level of text lines, RDI-CleverPage is in second place, and ECDP rarely oversegments text blocks.
- The under-segmentation error (USE) compares the number of under-segmented blocks to the number of ground-truth blocks. ECDP is more likely to merge text blocks thus it has a slightly larger USE than RLSA or RDI-CleverPage.
- The correct segmentation (CS) metric compares the number of correctly segmented blocks to the number of ground truth blocks. RLSA and RDI-CleverPage outperform ECDP.
- The missed segmentation error (MSE) compares the number of missed segments to the total number of ground truth blocks. ECDP has a lower MSE than both RLSA and RDI-CleverPage.
- The false alarm error (FA) compares the number of false alarms to the total number of ground truth blocks. RLSA has very large number of false alarms compared to ECDP, RDI-CleverPage is in second place.
- The overall block error rate ρ combines OSE, MSE, and USE and compares the result to the total number of ground truth blocks. The overall ECDP error is significantly less than RLSA and also lower than RDI-CleverPage.

Our comparison of RLSA, RDI-CleverPage, and ECDP with respect to classification performance uses four metrics: precision (Pr), recall (Rec), $F1$-measure ($F1$), and average text versus non-text accuracy (Acc). We applied these metrics to both pixel- and block-based results. The results, summarized in Table 12, show the superior classification performance of ECDP in 12 of the 14 metrics.

The remaining five experiments were conducted to examine the robustness of our ECDP system and reveal its limitations. We designed the experiments so that we could analyze the performance of ECDP on documents with content types that were not used in training or validating ECDP. In particular, from BCE-Arabic-v1, we used (1) 79 new pages of BCE-Arabic-v1 with text and charts, (2) 100 pages with text and diagrams, and (3) a few document images with complex multicolumn layouts. From the Hesham dataset, we used all the 60 book pages, 10 historical document pages, and 15 magazine pages. From the RDI dataset, we used all 109 book and magazine pages.

We found that ECDP can classify BCE-Arabic pages with chart content accurately, even locating text on the chart axes (Figs. 12, 13). For 79 documents with charts, we obtained average accuracy values of $A = 90.94$ and $F_1 = 90.19\%$. We found that ECDP handles bar charts (Fig. 12) better than line charts (Fig. 13). Straight lines in the chart frame or grid were often misclassified as text especially if the chart

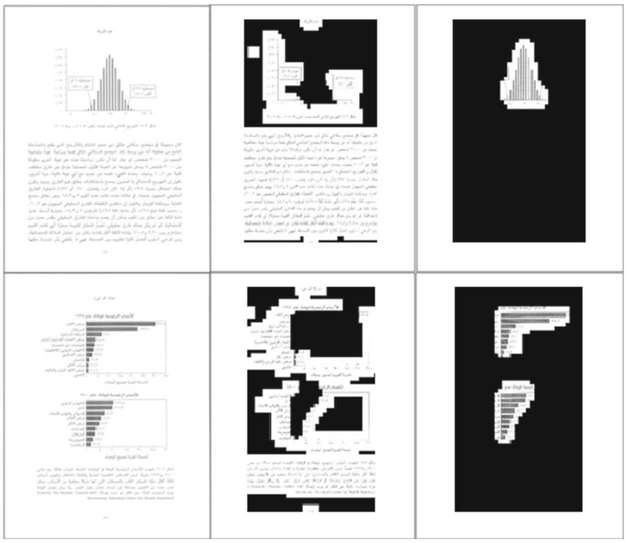


Fig. 12 ECDP performance on BCE-Arabic documents with bar charts. Text (center) and non-text (right) zones are detected in the two sample test images (left). Only the chart axes are mislabeled as text

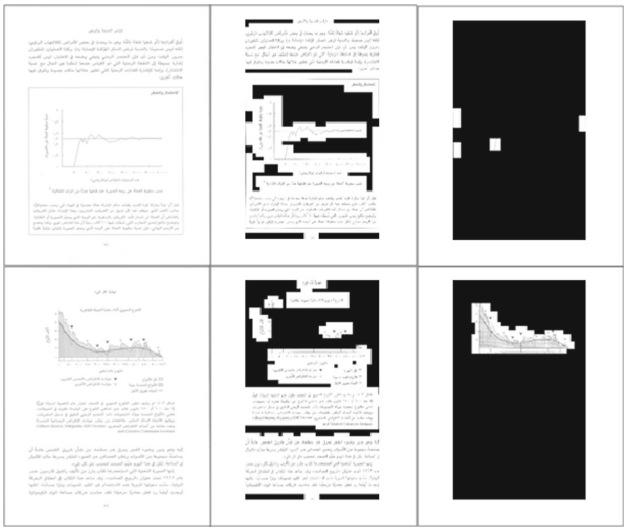


Fig. 13 ECDP performance on BCE-Arabic documents with line charts. Text (center) and non-text (right) zones are detected in the two sample test images (left). The diagram axes are mislabeled as text, as well as the function plot in the first example

background is plain. Similarly, line drawings may also be misinterpreted as containing text regions. Furthermore, we found that detected zones boundaries were not smooth since they were built using square image patches (e.g., the staircase pattern of the boundary of the non-text zones in Fig. 12).

Testing ECDP on the 100 pages of BCE-Arabic with text and diagrams revealed that the overall performance did not degrade much (Fig. 14). We obtained average accuracy values of $A = 87.89$ and $F_1 = 87.69\%$.

Testing ECDP on five pages of BCE-Arabic-v1 with multiple columns and complex layouts showed that this classification task was beyond the capability of ECDP (Fig. 15). The font size of the text in these pages was significantly smaller than the font size in the training data (Fig. 16). Resizing the image to four times its original size was not helpful since the pixel resolution of the characters in these images was poor. For ECDP to work on complex layouts, the input image has to be mosaicked, resized, resolution-enhanced and the model has to be modified to work on smaller patch sizes and also trained on inverse images (dark background and light text) and various font sizes.

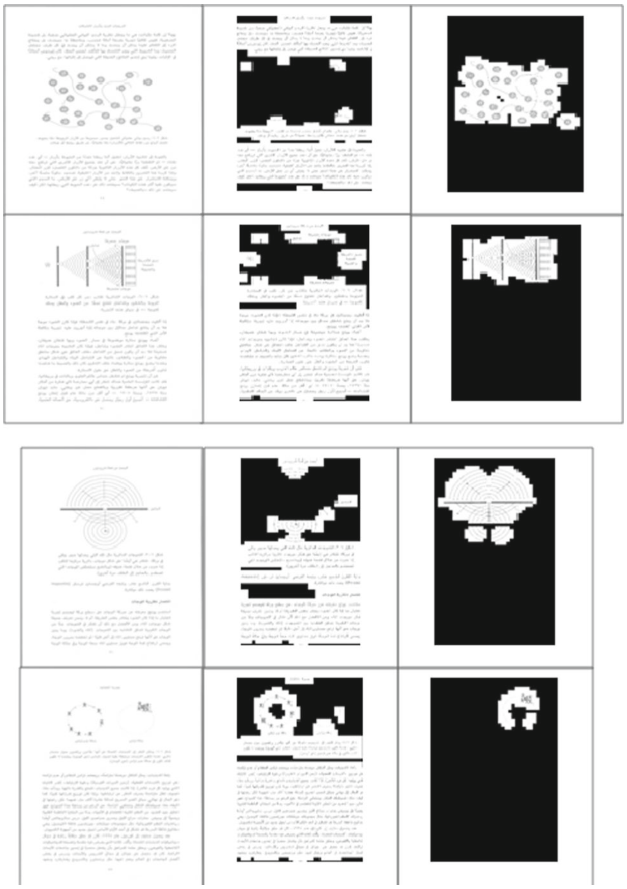


Fig. 14 ECDP performance on BCE-Arabic documents with diagrams. Text (center) and non-text (right) zones are mostly detected in the two top sample test images and partially in the bottom two (left)

ECDP achieved an average patch classification accuracy of 74.26% and average $F1$-measure of 76.08% when introduced to the Hesham dataset, and an average patch classification accuracy of 87.32% and average $F1$-measure of 89.4% when introduced to the RDI dataset. Hesham et al. [8] report higher performance numbers for their system on these datasets. A quantitative performance comparison between their system and ECDP, however, is not meaningful due to the use of (1) different ground-truth segmentations and (2) different segmentation evaluation metrics (e.g., Hesham et al. did not take into consideration segmentation errors of

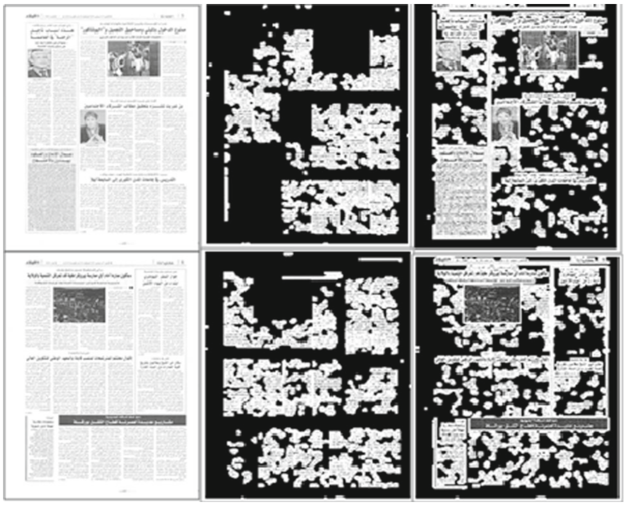


Fig. 15 ECDP did not perform well on documents with complex layouts: original image (left), detected text (center) and non-text (right) zones

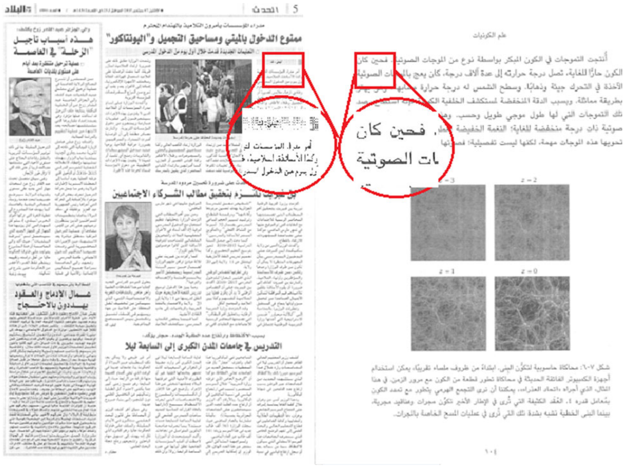


Fig. 16 Text patches of the complex layout document on the left have small fonts and look more like graphics patches (zoomed subimage, left), compared to the text patches in a book document image on the right for the same patch size (zoomed subimage, right)

region overlaps and splits). Visual inspection of ECDP results showed that the diacritic marks, dots that are associated with 15 out of the 28 characters in the Arabic alphabet, contributed to misclassification of patches (the system by Hesham et al. circumvents the problems diacritics cause by removing diacritics from further consideration at an early stage). Our error analysis furthermore revealed that the fonts and inter-line distances in the documents of Hesham and RDI datasets with low ECDP performance were significantly different from those of our BCE-Arabic training data. Adding documents with smaller inter-line distances and with a wider variety of font types and sizes to the training set of ECDP would likely boost its performance, especially on the Hesham dataset.

5 Discussion

This paper presents the first component of an end-to-end machine learning system for analyzing the layout of Arabic documents, specifically, scanned book pages in untagged raster-image PDF format. The results suggest that the proposed method has the potential to generalize well to the analysis of documents with a broad range of content.

The contributions in this paper can be summarized as follows:

- A machine learning approach for physical layout analysis of Arabic documents was proposed that uses an ensemble of support vector machine classifiers to categorize text and non-text image regions.
- The approach is unique in classifying small, overlapping images patches. Edge and Fourier descriptors, which work particularly well for analyzing Arabic script, are applied to the patches, and then neighboring patches of the same class are grouped into larger image zones.
- Re-classification turned out to be a successful strategy. Context information from overlapping and neighboring patches is used to check the potential need for re-classification of patch labels, before patches are grouped into homogeneous document zones.
- Accurate classification results were obtained on images from BCE-Arabic, a recently created dataset of Arabic document images that include rotated text and text embedded in graphics and figures.
- Reasonably accurate classification results were obtained on images from the RDI and Hesham datasets.
- The proposed method outperforms a traditional method (RLSA) and a state-of-the-art commercial product (RDI-CleverPage).

It is important to note that our ECDP system works on a patch level rather than pixel or connected-component level. It can thus exploit information in the vicinity of each pixel, its "context." Working with patches also enables our system to use connected-component edge/stroke features together with the pixel-based texture features. It avoids the drawbacks of methods based on grouping connected components, which are less likely to succeed with degraded document images where these components are broken. Moreover, there are far fewer patches than pixels or connected components, and so it is more time efficient to work on patches. Context information is also represented by the overlap of neighboring patches—this adds a small amount of useful redundancy, constructively contributing to the classification result.

Our choice to compare our ECDP system to the classical run-length smearing algorithm (RLSA) [15] was guided by the work of Shafait et al. [63]. The authors noted that RLSA was, at the time, the only algorithm that includes a block-classification step in addition to the segmentation step. To conduct a fair comparison to ECDP, we had to choose a method that performs both steps.

ECDP showed superior pixel-based and block-based performance compared to RLSA, except for the precision of non-text pixels and recall of text pixels. For these two metrics, both methods produced almost perfect pixel-based results, with RLSA winning by about two percentage points (Table 12) (RLSA $> 99\%$ vs. ECDP $> 97\%$). As a pixel-based method, RLSA misclassified fewer speckles of pixels (single pixels or very small collections) than the patch-based ECDP method. In particular, it misclassified fewer border-noise pixels as non-text rather than background (higher non-text precision) and fewer graphics pixels as text (higher text recall). For larger collections of pixels (and, even more so, blocks), ECDP outperforms RLSA since a patch-based approach forces all encountered pixels in the collection to have the same label as the patch label.

Comparing system performance is difficult due to the lack of benchmarking datasets that include Arabic documents in general, and “Arabic DLA benchmarking datasets” specifically, even in the form of a limited competition dataset. Until recently, before BCE-Arabic-v1 was published, the lack of a publicly available benchmark containing Arabic documents meant that researchers could only report their DLA system results on relatively small proprietary datasets (e.g., the RDI and Hesham datasets [8]) or small customized subsets they chose from public datasets matching the problem they address [3,4]. Now that BCE-Arabic-v1 is publicly available, we hope that our ECDP system can serve as a baseline system so that the performance of future systems (designed by others or ourselves) can be compared against a baseline.

In the PLA literature in general, and the learning-based PLA literature in particular, researchers have reported a wide variety of methods, datasets, evaluation metrics, and targets for the final system output. This is notable when both tasks of PLA are addressed, segmenting of the image into a set of non-overlapping homogeneous zones and labeling each zone according to its content class. System performance on the segmentation task only can be more readily compared to that of “classic algorithms.” Mao et al. [64], for example, compared their segmentation solution to the output of three classic segmentation algorithms (XY cut [10], Docstrum [12] and Voronoi [13]). Oyedotun et al. [65], who used a related approach to ours by extracting textural features from image segments by 4×4 sliding mask on the image regions, used a dataset of only 35 documents and classified features with a multi-layer perceptron neural net. The authors could not compare their results (84% average segment identification) against other PLA systems, so they implemented two traditional algorithms (the run-length smearing algorithm and adaptive k-means clustering) to compare the performance of their approach. The trend to compare a new system to classic algorithms might be acceptable for segmentation evaluation, but is less helpful when classification evaluation is also needed.

Finally, PLA solutions in the literature are typically not accompanied by published source code and reimplementing them is a non-trivial task, hindered by the fact that implementation details are often not published. We here, at least, were able to provide a performance comparison of our system to a state-of-the-art commercial product (RDI-CleverPage). By making our source code publicly available at http://www.cs.bu.edu/fac/betke/research/ECDP, we aim to contribute to a culture of code sharing in our community that might propel research forward.

6 Conclusion

This paper presented a solution to an important document analysis problem—how to make scanned PDF documents machine accessible. The proposed system ECDP performs physical layout analysis for Arabic scanned book pages in PDF format that contain text and non-text elements using a machine learning based method. ECDP classifies image patches using an SVM-ensemble trained on edge features and shape descriptors. It builds up the extracted text zones from the classified patches and recognizes the text stream with the Tesseract OCR package. The input document image is made machine accessible through processing of the two system output files: the XML hierarchy transcript file (layout description) and the recognized text file.

The system results are promising—accurate patch classification accuracy values were achieved, on average, for documents in three different datasets. Our experimental methodology proved to be successful. Testing the system on document samples with the content similar to the training examples showed particularly high performance. Testing ECDP on content of a nature previously not seen revealed good performance in detecting text zones, even if the text was embedded within graphics regions or rotated by 90 degrees. ECDP also succeeded in detecting some English text.

Our system needs some modification to address more complex layouts with low-resolution text, differing font types, font sizes, and text line distances, and a larger variety of non-text types. Thus, as a future work, we intend to proceed with further enhancements:

- Adjust the system to detect and process more challenging layouts and a wide variety of page contents like tables and line drawings by adding experts to the ensemble.
- Include training data with different layouts, including a wide variety of text properties and non-text elements.
- Try different classification approaches for performance comparison.

- Train different models on multi-lingual documents and build a language detection stage to obtain a language-independent solution.

In future work, we will also explore how transfer learning might be applied to solve the physical layout analysis of Arabic documents when a deep neural network has been pre-trained with English documents. To accomplish this, one must collect and annotate a larger dataset of Arabic document images. Our ultimate research goal is to arrive at a system that can be shown to work successfully for people with visual impairments.

Acknowledgements The authors would like to thank the RDI team for sharing their datasets and especially Shaimaa Samir for testing our data on the RDI Clever Page software. The authors acknowledge partial funding from the National Science Foundation (1337866, 1421943) (to M.B.) and the Cairo Initiative Scholarship Program (to R.E.).

References

1. Cattoni, R., Coianiz, T., Messelodi, S., Modena, C.M.: Geometric layout analysis techniques for document image understanding: a review. Technical Report TR9703-09, ITC-IRST, Trento, January 1998, 68 pages
2. Chen, K., Seuret, M., Wei, H., Liwicki, M., Hennebert, J., Ingold, R.: Ground truth model, tool, and dataset for layout analysis of historical documents. In: Proceedings of SPIE 9402, Document Recognition and Retrieval XXII, Feb. 2015, 10 pages
3. 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)
4. Bukhari, S.S., Breuel, T.M., Asi, A., El-Sana, J.: Layout analysis for Arabic historical document images using machine learning. In: International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 639–644 (2012)
5. Hadjar, K., Ingold, R.: Physical layout analysis of complex structured Arabic documents using artificial neural nets. In: International Workshop on Document Analysis Systems, pp. 170–178 (2004)
6. Saad, R.S.M., Elanwar, R.I., Abdel Kader, N.S., Mashali, S., Betke, M.: BCE-Arabic-v1 dataset: a step towards interpreting Arabic document images for people with visual impairments. In: ACM 9th Annual International Conference on Pervasive Technologies Related to Assistive Environments (PETRA'16), pp. 25–32, Corfu, June (2016)
7. BCE-Arabic Dataset: Publicly Available. http://www.cs.bu.edu/faculty/betke/BCE June (2016)
8. Hesham, A.M., Rashwan, M.A., Al-Barhamtoshy, H.M., Abdou, S.M., Badr, A.A., Farag, I.: Arabic document layout analysis. Pattern Anal. Appl. **20**, 1275–1287 (2017)
9. Proprietary Dataset of 109 Arabic Documents: Provided by RDI, The Engineering Compuany for Digital Systems Development. http://www.rdi-eg.com
10. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. Computer **7**(25), 10–22 (1992)
11. Baird, H.: Background structure in document images. Int. J. Pattern Recognit. Artif. Intell. **8**(5), 1013–1030 (1994)
12. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Recognit. Mach. Learn. (TPAMI) **15**(11), 1162–1173 (1993)
13. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. Comput. Vis. Image Underst. **70**(3), 370–382 (1998)
14. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Workshop on Document Analysis Systems, pp. 188–199, Princeton (2002)
15. Wong, K., Casey, R., Wahl, F.: Document analysis system. IBM J. Res. Dev. **26**(6), 647–656 (1982)
16. Jain, A.K., Yu, B.: Document representation and its application to page decomposition. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **20**(3), 294–308 (1988)
17. Fletcher, L.A., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **10**, 910–918 (1988)
18. Shafait, F., Hasan, A., Keysers, D., Breuel, T.M.: Layout analysis of Urdu document images. In: 10th International Multitopic Conference, pp. 293–298, Islamabad (2006)
19. Won, C.S.: Image extraction in digital documents. J. Electron. Imaging **17**(3), 033016 (2008)
20. Bukhari, S.S., Shafait, F., Breuel, T.M.: The IUPR dataset of camera-captured document images. In: International Workshop on Camera-Based Document Analysis and Recognition, pp. 164–171 (2012)
21. Lin, M.W., Tapamo, J.R., Ndovie, B.: A texture-based method for document segmentation and classification. S. Afr. Comput. J. **36**(1), 49–56 (2006)
22. Bukhari, S.S., Azawi, A., Ali, M.I., Shafait, F., Breuel, T.M.: Document image segmentation using discriminative learning over connected components. In: DAS'10 Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 183–190, Boston, June (2010)
23. Ye, X., Cheriet, M., Suen, C.Y.: A generic system to extract and clean handwritten data from business forms. In: 7th International Workshop on Frontiers in Handwriting Recognition, pp. 63–72 (2000)
24. Fan, W., Sun, J., Naoi, S.: Separation of text and background regions for high performance document image compression. In: Proceedings SPIE 9402, Document Recognition and Retrieval XXII, pp. 94020K1–9420K12, February (2015)
25. Le, V.P., Nayef, N., Visani, M., Ogier, J.M., De Tran, C.: Text and non-text segmentation based on connected component features. In: IEEE 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1096–1100, August (2015)
26. Rahman, A.F.R., Fairhurst, M.C.: Multiple classifier decision combination strategies for character recognition: a review. Int. J. Doc. Anal. Recognit. (IJDAR) **5**, 166–194 (2003)
27. Guyon, I., Haralick, R.M., Hull, J.J., Phillips, I.T.: Data sets for OCR and document image understanding research. In: Wang, H.B. (ed.) Handbook of Character Recognition and Document Image Analysis, pp. 779–799. World Scientific, Singapore (1997)
28. Taghva, K., Nartker, T., Borsack, J., Condit, A.: UNLV-ISRI document collection for research in OCR and information retrieval. In: IS&T/SPIE Conference on Document Recognition and Retrieval VII, San Jose, pp. 157–164, January (2000)
29. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition, pp. 181–188, Curitiba (2007)
30. Zelenika, D., Povh, J., Ženko, B.: Text detection in document images by machine learning algorithms. In: 9th International Conference on Computer Recognition Systems CORES 2015, pp. 169–179. Springer (2016)

31. 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)
32. 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 2007), vol. 2, pp. 1058–1062 (2007)
33. Bukhari, S.S., Shafait, F., Breuel, T.M.: Layout analysis of Arabic script documents. In: Margner, V., El Abed, H. (eds.) Guide to OCR for Arabic Scripts, pp. 35–53. Springer, London (2012)
34. Bloomberg, D.: Multiresolution morphological approach to document image analysis. In: 1st International Conference on Document Analysis and Recognition, pp. 963–971 (1991)
35. Breuel, T.M.: An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis. In: 7th International Conference on Document Analysis and Recognition, pp. 66–70 (2003)
36. Hadjar, K., Ingold, R.: Arabic newspaper page segmentation. In: 7th International Conference on Document Analysis and Recognition, pp. 895–899 (2003)
37. Capobianco, S., Marinai, S.: Text line extraction in handwritten historical documents. In: Italian Research Conference on Digital Libraries, pp. 68–79 (2017)
38. Pastor-Pellicer, J., Afzal, M.Z., Liwicki, M., Castro-Bleda, M.J.: Complete system for text line extraction using convolutional neural networks and watershed transform. In: 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, pp. 30–35 (2016)
39. Wick, C., Puppe, F.: Fully convolutional neural networks for page segmentation of historical document images. https://arxiv.org/pdf/1711.07695.pdf (2017), 6 pages
40. Hao, L., Gao, L., Yi, X., Tang, Z.: A table detection method for PDF documents based on convolutional neural networks. In: 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini (2016)
41. Meier, B., Stadelmann, T., Stampfli, J., Arnold, M., Cieliebak, M.: Fully convolutional neural networks for newspaper article segmentation. In: 14th International Conference on Document Analysis and Recognition (ICDAR), pp. 1–6 (2017)
42. Oliveira, D.A.B., Viana, P.M.: Fast CNN-based document layout analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1173–1180, Waikiki (2017)
43. Capobianco, S., Marinai, S.: Deep neural networks for record counting in historical handwritten documents. In: Pattern Recognition Letters (2017). https://doi.org/10.1016/j.patrec.2017.10.023
44. Rahman, M.M., Finin, T.: Understanding the logical and semantic structure of large documents. https://arxiv.org/pdf/1709.00770.pdf (2017), 10 pages
45. Afzal, M.Z., Capobiancot, S., Malik, M.I., Marinait, S., Breuel, T.M., Dengel, A., Liwicki, M.: DeepDocClassifier: document classification with deep convolutional neural network. In: 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1111–1115 (2015)
46. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 991–995, August (2015)
47. Kölsch, A., Afzal, M.Z., Ebbecke, M., Liwicki, M.: Real-time document image classification using deep CNN and extreme learning machines. https://arxiv.org/pdf/1711.05862.pdf (2017), 6 pages
48. Seuret, M., Fischer, A., Garz, A., Liwicki, M., Ingold, R.: Clustering historical documents based on the reconstruction error of autoencoders. In: ACM 3rd International Workshop on Historical Document Imaging and Processing, pp. 85–91 (2015)
49. 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)
50. 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)
51. 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)
52. Zhu, W., Chen, Q., Wei, C., Li, Z.: A segmentation algorithm based on image projection for complex text layout. In: American Institute of Physics (AIP) Conference Proceedings, vol. 1890, p. 1 (2017)
53. Ahn, B., Ryu, J., Koo, H.I., Cho, N.I.: Textline detection in degraded historical document images. EURASIP J. Image Video Process. **82**, 1–13 (2017)
54. Zhang, X., Duan, L., Ma, L., Wu, J.: Text extraction for historical Tibetan document images based on connected component analysis and corner point detection. In: Yang, J., et al. (eds.) Computer Vision. CCCV 2017. Communications in Computer and Information Science, vol. 772, pp. 545–555. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-7302-1_45
55. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Aletheia—an advanced document layout and text ground-truthing system for production environments. In: IEEE International Conference on Document Analysis and Recognition (ICDAR), pp. 48–52, September (2011)
56. 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)
57. Kavitha, A.S., Shivakumara, P., Kumar, G.H., Lu, T.: Text segmentation in degraded historical document images. Egypt. Inf. J. **17**, 189–197 (2016)
58. Wang, Y., Phillips, I.T., Haralick, R.M.: A study on the document zone content classification problem. In: International Workshop on Document Analysis Systems, pp. 212–223 (2002)
59. Baechler, M., Bloechle, J.-L., Ingold, R.L.: Semi-automatic annotation tool for medieval manuscripts. In: 2010 IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 182–187 (2010)
60. Deivalakshmi, S., Palanisamy, P., Vishwanathan, G.: A novel method for text and non-text segmentation in document images. In: 2013 IEEE International Conference on Communications and Signal Processing (ICCSP), pp. 255–259 (2013)
61. Tahmasbi, A., Saki, F., Shokouhi, S.B.: Classification of benign and malignant masses based on Zernike moments. Comput. Biol. Med. **41**(8), 726–735 (2011)
62. RDI CleverPage: Document layout analysis software by RDI, The Engineering Compuany for Digital Systems Development. http://www.rdi-eg.com
63. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 941–954 (2008)
64. Mao, S., Kanungo, T.: Empirical performance evaluation methodology and its application to page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **23**(3), 242–256 (2001)
65. Oyedotun, O.K., Khashman, A.: Document segmentation using textural features summarization and feedforward neural network. Appl. Intell. **45**, 1–15 (2016)