ASAR 2018 Layout Analysis Challenge: Using Random Forests to Analyze Scanned Arabic Books

Rana S.M. Saad*, Randa Elanwar*, N.S. Abdel Kader†, Samia Mashali*, and Margrit Betke‡
*Department of Computers and Systems, Electronics Research Institute, Cairo, Egypt
†Department of Electronics and Communications Engineering, Cairo University, Egypt
‡Department of Computer Science, Boston University, USA
Emails:rana,randa.elanwar,samia@eri.sci.eg; nemat2000@hotmail.com; betke@bu.edu

***Abstract*—Physical Layout Analysis (PLA) is a necessary step to recognize the contents of a digital document. PLA includes segmenting the document image and identifying the content type of the segments. PLA for digitized Arabic documents is challenging due to the nature of the Arabic script. In this paper, we introduce a PLA system for Arabic documents that were digitized by scanning. Our system RFAAD, short for "Random Forests for Analyzing Arabic Documents," starts with morphological preprocessing of the digitized hard copy and then extracts geometrical, shape, and context features to identify the connected components (CC) of the digital image as containing text or non-text. Random forests are trained using the first dataset release of a large data collection project, BCE-Arabic-v1 [22]. Our system shows strong performance on BCE data in terms of CC classification accuracy and F1-score (97.5% and 97.7% respectively). When evaluated on datasets by other researchers [2], [11], RFAAD also performs well. Moreover, RFAAD shows moderately strong performance when applied to the most challenging layouts of the benchmarking dataset of the ASAR 2018 competition PLA-SAB.[1] The performance of RFAAD suggests that our work, with some modifications, has the potential to solve other open problems in the document analysis area and attain a relatively high degree of generalization.**

1. Introduction

There is a recent increase in the number of digitized Arabic materials of all forms on the Internet. This can be attributed to projects for archiving out-of-print documents (e.g., the Arabic Collection Online digital library) and preserving ancient heritage (e.g., the Islamic Heritage Project). An additional reason is the rapid increase of shared personal document collections (images, PDF files, etc.).

Unlike English and Chinese, the Arabic digital content on the Internet is still limited, in spite of the fact that Arabic is ranked as the 5th most spoken language world-wide [22], spoken by over 400 million people [2]. Moreover, the Arabic alphabet is the basis of many languages like Urdu, Persian, Hausa, Swahili, Tigrigna, Pashto, Wolof, Kurdish, Jawi, Berber, Amharic, and Pular.

Digital documents are created either by word processing software, which saves them in tagged-PDF format, or by digitizing a hard copy using a scanner or a camera, which saves them in raster-PDF format. PDFs with tags have a full hidden transcription of the file content of text script and images. On the other hand, raster-PDFs have no tags and are composed of images representing the document pages. As discussed by AlMasoud and Al-Khalifa [1], the latter type of PDFs is entirely untagged with no information about its content, thus cannot be retrieved or searched. Since all digitization projects produce bulks of raster PDFs that need expert help to associate metadata for search and retrieval, proposing a solution to help automate the information extraction from a document image is much appreciated.

The Arabic language has a specific nature that adds difficulty to the challenge of analyzing a document image. Arabic words are composed of one or more connected parts. Each part of the Arabic word (PAW) can represent 1 to 5 characters cursively written unlike English. The existence of diacritics, dots, and ligatures complicate analysis and recognition [15]. Different handwriting styles and decorative printed fonts pose additional difficulty to the available document analysis and recognition approaches. Thus, lower performance when compared to English documents usually occurs. It is recommended to build customized solutions for Arabic documents rather than using one that is language-independent. However, building such a solution will face many obstacles – most importantly, the fact that large research datasets of Arabic documents are not publicly available.

Generally, successful document layout analysis (DLA) is a key step towards the understanding of a document image and the recognition of its content. A full DLA system is composed of two stages involving physical and logical layout analysis. Physical layout analysis (PLA) discovers the identity of the structural components of a document as text, images, charts, etc. Logical layout analysis defines the functional role of the textual components within the page such as header, footer, headings, main paragraph, etc.

In this paper, we focus on the PLA stage only. PLA can

1. http://www.cs.bu.edu/faculty/betke/ASARLayoutAnalysisCompetition

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

be achieved via a top-down approach by first segmenting the page image into blocks followed by block classification, or may result from a bottom-up approach by classifying the initial document primitives (pixels, connected components 'CC', or patches), followed by block formation.

Top-down PLA approaches suffer mainly from inaccurate segmentation results, as been pointed out by Shafait et al. [23]. They are not immune to noise or document skewing. On the other hand, bottom-up PLA approaches suffer from high computational cost, which was discussed by Marinai [18]. The tradeoff between accuracy and cost is partly solved by considering CCs or superpixels as analysis primitives instead of pixels.

In this paper, we perform PLA by following the bottom-up approach. Our system RFAAD classifies features extracted from CCs to identify their type. Then blocks are built and their bounding boxes extracted, and, finally a tagged transcription of a raster-PDF is generated.

Using the geometric features of connected components to analyze the content of a document image has been used previously in conjunction with various classifier types. Liang et al. [17] used decision trees to classify the document components into eight classes depending on the width and height of the CCs, whereas Zelenika et al. [27] used the CCs shape and context features in evaluating the performance of different classifiers. Cohen et al. [6] used a conditional random field in categorizing document CCs into text and non-text based on their dimensions and stroke width. Zagoris et al. [26] merged the labeled CCs to calculate the standard deviation of document structure elements and used a SVM to classify them. Le et al. [16] used Hu moments as shape descriptors of the labeled CCs in addition to some context features to be classified by an Adaboost decision tree.

PLA solutions have been mostly introduced for Latin-alphabet languages, Chinese, and Hindi, but not Arabic. Also, most Arabic PLA systems in the literature address analyzing handwritten documents (historical manuscripts) [5], [6] rather than machine-printed documents such as pages from modern books and newspapers [2], [11].

In this paper, we propose a machine-learning-based solution that performs physical layout analysis to scanned modern book pages using a Random Forests classifier. Different datasets of documents are used to evaluate the proposed system. It shows good results for our self-collected, publicly available Arabic document dataset BCE–Arabic–v1, as well as datasets collected by other researchers, in particular, the dataset RDI and the dataset by Hesham et al. [2], [11]. Additionally, a good record is achieved for dataset of the ASAR 2018 Layout Analysis Challenge PLA-SAB [8].

2. System Description

RFAAD is a learning-based system for analyzing the physical layout of scanned Arabic documents. RFAAD follows the bottom-up approach of classifying document image connected components. Elementary preprocessing in the form of binarization, noise removal, and morphological operations are applied. Descriptive structural features are extracted from CCs of the processed image, which are used further in training a Random Forest (RF) classifier. In the test stage, the feature vectors from the processed image are classified by the RF, blocks are built, and a transcript is created for the document.

2.1. Preprocessing

Initially, the input gray scale image experience binarization using Otsu threshold [20], then a median filter is applied to the binarized image for noise removal. Any residual border noise is further suppressed.

We noticed from preliminary experiments that the binarization process produces a lot of broken components that are similar in shape and size to diacritics associated with the Arabic alphabet. Thus, classifier confusion is highly expected.

One solution is to use morphological operations to combine these broken components into larger components. This operation will lead to significant size differences between text and non-text components as the latter will appear much larger.

A sequence of morphological operations is performed to the binarized image (Fig. 1):

1) Extract Canny edges of the binarized image,
2) Perform *Closing* with a 3×3 structure element,
3) Perform *Dilation* with a 3×3 structure element, and finally
4) Fill the dilated image with a 3×3 structure element.

2.2. Connected Component Analysis

After preprocessing, structural features are extracted from the connected components, an approach inspired by Le et al. [16]:

Normalized CC Centers

It has been observed that most text and noise components are usually located near the document borders unlike non-text components.

$$x_{\text{normalized}} = \frac{CC\, x_{center}}{\text{Document width}} \tag{1}$$

$$y_{\text{normalized}} = \frac{CC\, y_{center}}{\text{Document height}} \tag{2}$$

Normalized width and height of CCs bounding boxes

The dimensions of CCs are expected to differentiate text from non-text components as non-text dimensions are usually larger.

$$\text{Width}_{\text{normalized}} = \frac{\text{CC width}}{\text{Document width}} \tag{3}$$

$$\text{Height}_{\text{normalized}} = \frac{\text{CC height}}{\text{Document height}} \tag{4}$$

Elongation

Elongation is defined as the ratio between CC width and

Figure 1. Preprocessing operations applied to an input document image.

height. It can be used to discriminate text from arbitrary-shaped lines.

$$\text{Elongation} = \frac{\min(width, height)}{\max(width, height)} \tag{5}$$

Solidity
Solidity is defined as the ratio between the foreground pixels and area of a CC. It can be used to discriminate text from images.

$$\text{Solidity} = \frac{No.\ of\ pixels\ in\ CC}{Area\ of\ CC\ bounding\ box} \tag{6}$$

Log normal distribution of height
This distribution is useful for differentiating large size fonts from non-text components:

$$\text{Log}_\text{n}(\text{height}) = \frac{\text{height}}{\sigma\sqrt{2\pi}} \times \exp \frac{-(\ln(\frac{1}{\text{height}} - u))^2}{2\sigma^2} \tag{7}$$

The mean u and standard variation σ used are based on the values for the CCs in the training dataset. **Hu moments**
Hu moments [12] can describe the CCs shape. They are invariant to rotation or reflection.
Mean and standard deviation of the stroke width of text
The text that occurs in a single page usually has approximately the same stroke width. The stroke width (SW) is defined as the perimeter of the CC edges over the number of its pixels. The edges of a CC are detected by Canny edge detection method.

$$SW = \frac{CC_{perimeter}}{No.\ of\ pixels} \tag{8}$$

Nearest neighbor features
For each CC, ten nearest-neighbor CCs are found. The mean component height $H_{neighbors}$, width $W_{neighbors}$, and stroke width $SW_{neighbors}$ of these neighbors are calculated:

$$W_{mean} = \frac{image_{width}}{mean(W_{neighbors})} \tag{9}$$

$$H_{mean} = \frac{image_{height}}{mean(H_{neighbors})} \tag{10}$$

$$SW_{mean} = \frac{mean_{SW}}{mean(SW_{neighbors})} \tag{11}$$

The extracted features are further normalized to zero mean and unity standard deviation so that their values lie betweem -1 and 1.

2.3. Random Forests Classifier

Researchers have used different types of classifiers for document layout analysis tasks such as neural networks (NN) [5], Support Vector Machines (SVMs) [25], decision trees [24], and Random Forests [7].

The Random Forest (RF) method is based on a classifier ensemble [4] and has been shown to outperform decision tree classifiers [3]. The RF model has been used to solve various image analysis tasks, for example, gesture classification [13]. It was found that, in some applications, RFs can perform as well as SVMs (e.g., classification in remote sensing [21]) or outperform SVMs (e.g., layout analysis and text detection [9]). Fang et al. [9] used RFs to detect the rows and columns of tables and determine whether those tables have only horizontal guiding lines, have both vertical and horizontal lines, or do not have any guiding lines.

Karimian et al. [14] suggested that the RF model is a good classifier due to its performance stability in the presence of noise, its fast learning procedure, and its ability to estimate missing information. Oshiro et al. [19] also pointed out its ability to handle large datasets.

In this paper, we investigate RF performance on classifying the connected components of Arabic documents as text or non-text. The next section, Section 3, presents our parameter tuning and validation experiments. Section 4 reports the evaluation experiments on six datasets including the dataset that was provided by the ASAR 2018 Layout Analysis Competition PLA-SAB 2018 [8].

3. Validation Experiments and Results

RFAAD is trained and tested using the BCE-Arabic–v1 dataset [22]. The BCE–v1 is the first publicly-available dataset of Arabic document images that is constructed mainly for layout analysis tasks. Images with text and image elements were selected (383 images) and are partitioned into training, validation, and test sets (60:20:20 respectively).

RFAAD was implemented using the Java-based WEKA-library [10] and run on a core i7 PC with 8 GB RAM.

In the validation experiments, two parameters were tuned: the number of trees, and the number of features per tree. Our experiments were inspired by the work of Joshi et al. [13]. For setting the number of features per tree (FV), the number of trees (N) was fixed at 100 trees. We experimented with values for FV as shown in Table 1.

TABLE 1. TUNING THE NUMBER OF FEATURES PER TREE USING THE VALIDATION DATASET, INVESTIGATING THE EFFECT ON THE AVERAGE CC CLASSIFICATION ACCURACY

# features	Classification Acc.
$\sqrt{FV}$	97.29%
$0.5 * \sqrt{FV}$	97.91%
FV	95.65%
$Log(FV)/Log(2)$	97.29%

Oshiro et al. [19] pointed out that using a large number of trees has a positive effect on the RF classification performance. Several values were investigated for the best feature vector length as shown in Table 2.

TABLE 2. TUNING THE NUMBER OF TREES USING THE VALIDATION DATASET, INVESTIGATING THE EFFECT ON THE AVERAGE CC CLASSIFICATION ACCURACY

# trees	Classification Acc.	# trees	Classification Acc.
1	93.2%	400	98.06%
50	97.93%	450	98.07%
100	97.91%	500	98.04%
150	97.93%	550	98.04%
200	97.95%	600	98.05%
250	97.97%	650	98.02%
300	97.98%	700	98.02%
350	98.03%	750	98.03%

The highest classification accuracy on the validation dataset is obtained with a 450-tree RF model. The best model results for BCE-V1 validation dataset are 97.9% as average CC classification accuracy and 97.8% as average F1 measure. Figure 2 shows a visual representation of the CC classification results of the best model on two sample book page images.

4. Evaluation and Benchmarking

We used six datasets to evaluate RFAAD. The first dataset, BCE-V1 [22] was published by the authors of this paper in 2016. We also evaluate RFAAD on datasets collected by others.

4.1. Evaluation using BCE-V1 test set

Introducing the BCE-V1 test set to the RFAAD model that yielded the best performance results in our validation experiments, we obtained a 97.5% average CC classification accuracy and a 97.7% average F1 measure. The first row in Table 3 shows detailed results for RFAAD on BCE–V1 for both text and non-text classes.

Figure 2. Examples of RFAAD performance on two images in the validation dataset. The green bounding boxes represent the non-text CC correctly classified as non-text. The blue bounding boxes represent the text CC correctly classified as text. The red bounding boxes represent the text CC mistaken for non-text. The magenta bounding boxes represent the non-text CC misclassified as text.

TABLE 3. CLASSIFICATION PERFORMANCE (%) OF RFAAD WITH RESPECT TO THE METRICS PRECISION (PR), RECALL (REC), F1-MEASURE (F1) AND AVERAGE TEXT VERSUS NON-TEXT ACCURACY (ACC) OVER SIX DIFFERENT TEST DATASETS

	Text			Non-text			Avg.
	Pr	Rec	F1	Pr	Rec	F1	CCAcc.
BCE-V1	99.8	97.6	98.7	63.1	94.5	75.7	97.49
RDI	99.5	91.6	95.4	18.9	79.6	30.6	91.33
Hesham et al. [11]	94.2	98.4	96.3	54.5	23.7	33.0	92.90
ASAR2018-A	74.4	99.9	85.3	97.9	15.7	27.1	75.52
ASAR2018-B	80.3	99.1	88.7	89.2	23.0	36.5	80.88
ASAR2018-C	99.4	95.6	97.4	69.6	94.4	80.1	95.47

4.2. Evaluation on datasets collected by other researchers

The state-of-the art is the work by Hesham et al. [11] for analyzing the layout of Arabic documents from modern books, magazines, and degraded historical books. We could not compare our segmentation results to theirs directly as they used different evaluation metrics for segmentation, and we also could not compare our classification results with theirs since they worked on a block level for classification.

As an alternative, we used the data collected by Hesham et al. [11] to test the robustness of our system. We used two of their datasets: (1) the 85 pages from books, magazines, and historical documents, and (2) the RDI dataset, which contains 109 scanned pages from Arabic books and magazines. The best model for RFAAD obtained 92.9% and 91.3% average CC classification accuracy and 91.6% and 93.8% average F1 measure for the two datasets, respectively. Table 3 shows detailed results for both text and non-text classes.

Some error cases are shown in Figure 3. In the dataset by Hesham et al., the authors used Savola's method for

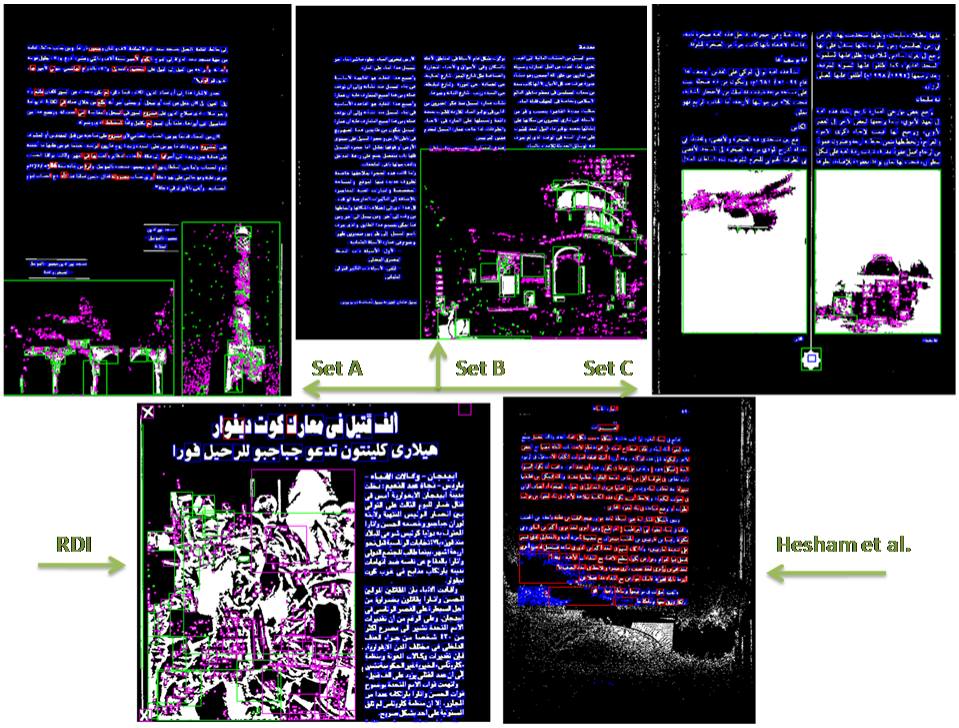

Figure 3. RFAAD results for five different test sets (see the caption of Fig. 2 for an explanation of the color scheme).

binarization while we used Otsu's method. The performance difference between the two algorithms significantly affected our results as shown in Figure 4. In the RDI dataset, the errors are mainly due to having different font sizes (unusual sizes are misclassified as non-text CC 'red') and small non-text components that could not be masked in the preprocessing stage (misclassified as text diacritics 'magenta'). More sophisticated preprocessing will resolve most of these cases.

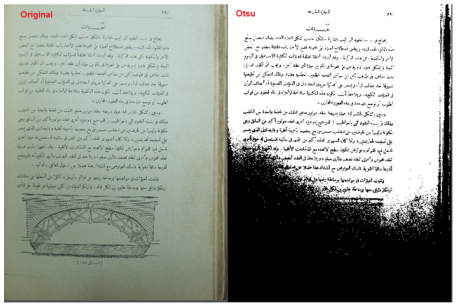

Figure 4. Performance differences in binarization methods.

4.3. Evaluation using ASAR 2018 layout analysis challenge dataset

The ASAR2018 layout analysis benchmarking dataset is composed of 90 images in 3 equal sets (A, B and C) according to layout shape and page content.

Introducing the ASAR 2018 challenge benchmarking dataset to RFAAD, we obtained 75.52%, 80.88%, and 95.47% as average CC classification accuracies for sets A, B, and C, respectively, and 68.5%, 76.2%, and 95.8% as respective average F1 measures. Table 3 shows detailed results for both text and non-text classes.

Some error cases are shown in Figure 3. As mentioned before, the errors are mainly due to small non-text components that could not be masked and therefore need to be handled with better preprocessing.

For the competition, segmentation evaluation was performed. Block building was performed by grouping adjacent same-class CCs, and segmentation evaluation was performed by comparison against the dataset ground truth and calculating the over-segmentation error (OSE), under-segmentation error (USE), missed-segmentation error (MSE), false alarm error (FA), correct-segmentation (CS), and the overall block error rate (ρ) [8]. We also calculated the average black pixel rate (AvgBPR), which represents the number of black pixels contained in segmented blocks compared to the corresponding blocks in the ground-truth image.

The respective RFAAD segmentation results for ASAR 2018 sets A, B, and C are: AvgBPR = 86.5, 90, and 78.66%, OSE = 1.37, 1.36, and 0.66, USE = 1, 1.68, and 0.59, MSE = 0.43, 1.07, and 1.45, FA = 2.5, 3.5, and 3.6, CS = 10.1, 13.25, and 9.4, ρ = 2.8, 4.1, and 2.7. RFAAD classification results on both pixel and block levels are shown in Table 4.

TABLE 4. CLASSIFICATION PERFORMANCE OF RFAAD OVER ASAR2018 BENCHMARKING DATASETS

	Pixels (%)				Blocks (%)			
	Pr	**Rec**	**F1**	**Acc.**	**Pr**	**Rec**	**F1**	**Acc.**
Set A	61.8	98.2	66.3	69.4	81.4	86.2	82.3	74.6
Set B	50.3	97.3	55.91	58.83	79.08	81.43	78.87	71.39
Set C	63.75	96.17	70.06	70.57	75.1	82.9	77	68.6

5. Conclusion

RFAAD is a CC-based system that performs PLA for scanned Arabic documents using a Random Forest Classifier. The method is invariant to document rotation and skew. It can perform well on both single and multi column layouts. RFAAD showed generally good performance. Errors occurred when preprocessing (binarization) affected the image resolution. Also, handling different font sizes is still a challenge that will need to be addressed in future work with a font detection methodology.

Acknowledgments

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] A. M. AlMasoud and H. S. Al-Khalifa. Investigating accessibility problems of arabic pdf documents. In *IEEE Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, pages 1–5, October 2013.

[2] 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):30–37, 2012.

[3] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180, 2007.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana. Layout analysis for Arabic historical document images using machine learning. In *2012 IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 639–644, 2012.

[6] R. Cohen, A. Asi, K. Kedem, and J. El-Sana I. Dinstein. Robust text and drawing segmentation algorithm for historical documents. In *ACM 2nd International Workshop on Historical Document Imaging and Processing*, pages 110–117, 2013.

[7] A. Corbelli, L. Baraldi, C. Grana, and R. Cucchiara. Historical document digitization through layout analysis and deep content classification. In *23rd International Conference on Pattern Recognition (ICPR)*, pages 4077–4082, December 2016.

[8] R. Elanwar and M. Betke. The ASAR 2018 Competition on physical layout analysis of scanned Arabic books (PLA-SAB 2018). In *2nd IEEE International Workshop on Arabic and derived Script Analysis and Recognition (ASAR 2018), London, March 2018.*, pages 1–6.

[9] J. Fang, P. Mitra, Z. Tang, and C.L. Giles. Table header detection and classification. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 599–605, July 2012.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[11] A. M. Hesham, M. A. Rashwan, H. M. Al-Barhamtoshy, S. M. Abdou, A. A. Badr, and I. Farag. Arabic document layout analysis. *Communications in Computer and Information Science*, 20(4):1275–1287, 2017.

[12] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[13] A. Joshi, C. Monnier, M. Betke, and S. Sclaroff. A random forest approach to segmenting and classifying gestures. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7, May 2015.

[14] F. Karimian and S.M. Babamir. Evaluation of classifiers in software fault-proneness prediction. *Journal of AI and Data Mining*, 5(2):149–167, 2017.

[15] M. S. Khorsheed. Off-line arabic character recognition a review. *Pattern analysis & applications*, 5(1):31–45, 2002.

[16] V. P. Le, M. Nayef, M. Visani, J. M. Ogier, and C. D. Tran. Text and non-text segmentation using connected component-based features. In *IEEE International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[17] J. Liang, R. Haralick, J. Ha, and I. T. Phillips. Document zone classification using sizes of connected components. *Electronic Imaging: Science & Technology. International Society for Optics and Photonics*, pages 150–157, 1996.

[18] S. Marinai. Learning algorithms for document layout analysis. In *Handbook of Statistics: Machine Learning: Theory and Applications*, chapter 31, pages 401–419. 2013.

[19] T.M. Oshiro, P.S. Perez, and J.A. Baranauskas. How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168, Springer, Berlin, Heidelberg, July 2012.

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

[21] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[22] R. S. M. Saad, R. I. Elanwar, N. S. Abdel Kader, S. Mashali, and M. Betke. 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)*, pages 25–32, Corfu, Greece, June 2016.

[23] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.

[24] Y. Wang, I. T. Phillips, and R. M. Haralick. A study on the document zone content classification problem. In *International Workshop on Document Analysis Systems*, pages 212–223, 2002.

[25] H. Wei, M. Baechler, F. Slimane, and R. Ingold. Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents. In *2013 IEEE 12th Document Analysis and Recognition (ICDAR)*, pages 1220–1224, August 2013.

[26] K. Zagoris and N. Papamarkos. Text extraction using document structure features and support vector machines. In *11th IASTED International Conference on Computer Graphics and Imaging*, pages 88–91, 2010.

[27] D. Zelenika, J. Povh, and B. Zenko. Text detection in document images by machine learning algorithms. In *9th International Conference on Computer Recognition Systems (CORES2015)*, pages 169–179, 2016.