Deep Arabic Document Layout Analysis

Ibrahim M. Amer, Salma Hamdy and Mostafa G. M. Mostafa

This is the accepted version of a conference paper published by IEEE.


© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Abstract—Document layout analysis (DLA) is an essential step for Optical Character Recognition Systems (OCR). The text of the document fed to the OCR must be extracted first and isolated from images if exist. The DLA task is difficult as there is no fixed layout for all documents, but instead, there are several layouts. There are various approaches for DLA for various different languages. In this paper, some of the previous techniques used in this field will be listed and then we will discuss the proposed method that depends on deep learning for documents’ text localization. We used Arabic Printed Text Image database (APTI [19]), ImageNet [18] and a dataset collected from different Arabic newspapers for training and evaluation.

Keywords—Optical Character Recognition, Document Layout Analysis, Font Type Recognition, Font Size Recognition, Deep Learning, Deep Convolutional Neural Networks (D-CNN), Transfer Learning (TL).

I. INTRODUCTION

Document Layout Analysis is an important step for OCR, document management systems, document archiving systems and more. OCR systems recognize printed or handwritten text images, these images must contain text only and if the document contains text mixed with photos, graphs shapes or halftones; this will result in a negative effect on recognition accuracy. Thus, DLA is a crucial step before OCR. DLA is a method of defining and recognizing the structure of the document or it is the way of categorizing the important regions in the document. It includes separating its text components from non-text to feed them directly to the OCR, identifying the title of the document if any, etc. Various font types and sizes are also factors that can be used in document analysis, because if the font type/size provided to the OCR is different from what the OCR waits, the results might not be precise. Reading order determination (right to left or left to right) is also a necessary factor to enhance the DLA and produce reliable results.

Document layout analysis has mainly two approaches. The first is the bottom-up approach, which is concerned with parsing the document in an iterative way. Methods based on bottom-up approach are concerned with grouping connected regions into words then into text lines and text blocks. The second approach is the top-down approach which is implied from its name that the document is segmented into columns first (relied on white space for example), segmenting its lines then finally, segmenting its words. Several algorithms and techniques are recently introduced for the DLA to documents written in English, but Arabic image documents did not get much attention compared with other languages.

This paper is organized as follows; section II summarizes some of the related works of Arabic document layout analysis. Section III shows the proposed technique. Finally, section IV discusses the achieved results and datasets used.

II. RELATED WORK

There are few papers that address the document analysis for Arabic documents. This section discusses some of them.

Syed Saqib Bukhari et. al [1] proposed a technique for text lines extraction as well as determining reading order from scanned documents written in various languages and different styles. The processing flow for their system has four main steps: Binarization, Text and Non-text segmentation, Text-line Detection and Reading Order Determination. For binarization step, they used Otsu [2] and Sauvola [3] methods. The paper presents an improvement for Bloomberg method in [6] for text and non-text segmentation. Bloomberg’s method does not work well for non-text elements such as drawings. Syed Saqib Bukhari et.al. proposed two modifications in [7] to deal with non-text elements. They used ridge-based line detection technique for text line detection, which is robust against noise, skew; also, it can be used for documents with different languages. Finally, the order of the text lines is detected and determined according to the reading flow (left to right or right to left). They perform the reading order by applying some ordering criteria according to Arabic reading order (right to left).

Amany, Mohsen et. al in [8] proposed a method for Arabic layout analysis. First, Sauvola binarization algorithm [3] is used for document binarization. Then, the images are cleaned up from noise induced by scanning process using
median filters to reduce salt and pepper noise and Gaussian smoothing filter to clear and eliminate both Gaussian and marginal noise. After that, the document is deskewed using Radon transform to correct its skew that might be induced by the scanning process. Next, the document is segmented into zones and then each zone is classified into text or non-text using Support Vector Machine (SVM) after extraction of some textual features. Finally, each zone is segmented into lines by clustering connected components into lines and words via clustering the spaces between words and between standalone characters of the same word using k-means. The system has been tested and evaluated on Arabic historical documents with different font types/sizes.

Syed Saqib Bukhari et al in [9] presented a method for side-notes extraction in Arabic manuscripts. Feature extraction is relied mainly on connected component analysis. Every connected component is re-sized to size 64x64. For each window, normalized height, foreground area, relative distance and orientation is calculated to form a feature vector of size (4 + 64x64) including the window itself. The surrounding context area of the connected component is also included in the feature vector. The final feature vector size is 8192. Finally, the side notes are classified using SVM. F-measure for main-body text segmentation is 95.02% and 94.68% for the side notes text.

III. PROPOSED METHOD

In this section, a method for Arabic DLA using deep learning is proposed. Given a document image, the target is to reveal, localize and isolate text regions from non-text ones. A Deep Convolutional Neural Network (D-CNN) is used to classify document’s regions (zones) into text or non-text. Given a document image, the document skew is corrected first then the document is pre-processed using Adaptive Run Length Smoothing Algorithm (ARLSA [10]) to merge adjacent pixels together to form a block/zone. This zone can be classified later into text or non-text. Then, after applying the ARLSA algorithm, all the connected components are identified and localized using connected component analysis. After that, each block/zone is classified using two techniques: 1) Zone Based Classification and 2) Patch Based Classification, more specific details will be mentioned in Classification subsection. Finally, each text zone is segmented into lines and words. The used dataset was collected from Arabic newspapers (Al-Sharq [11], Riyadh [12] and Al-Youm [13]). Datasets will be mentioned in detail in the Results and Discussion section. In this section, each component of the proposed system will be discussed and described in details. Proposed system architecture is shown in Fig. 1.

![Proposed System Architecture](image)

1) Pre-processing:
First, all documents are skew corrected using fast Hough transform with block adjacency graph (BAG) [5] and binarized using Bradley’s adaptive thresholding method [4] using a neighborhood of size 1/8th of the image size. Then, ARLSA is applied to the binarized image. ARLSA is used to segment the document image into its building blocks (text and non-text regions) by grouping adjacent pixels together to form a block/zone that can be fed to the classifier later in order to classify this zone as text or non-text. ARLSA is better than RLSA [14] because it can group in-homogeneous components and can work with documents written with variable font size. The effect of ARLSA or RLSA to the document image is similar to dilating the image using a structuring element but with applying some constraints as shown in Fig. 2. After that, bounding boxes of all segmented zones is identified using connected component analysis [15] [16] as in Fig. 3.
2) **Classification:** Since the dataset is manually collected from newspapers available online in PDF form, labeling such a dataset to be used with a deep learning technique consumes time and requires huge efforts, especially that DL techniques, are categorized as data hungry techniques, which require huge data for training. Transfer learning (TL) [17] concept is extremely helpful and useful in such cases. Transfer learning or knowledge transfer can be applied when having a classification task and a domain with sufficient training data called the source domain $D_S$, but there is no enough training data in another domain of interest called the target domain $D_T$, where the latter domain may have a different feature space ($FS$). Self-driving cars for example can be trained using the approach of TL by preparing a model that can segment and recognize cars, trucks and pedestrians in a computer simulated game; then, the model can be adapted to work with real images captured from the streets. In such cases, knowledge transfer if done effectively would incredibly enhance the performance of learning by avoiding much expensive data labeling efforts. Given a classification task in one domain with a sufficient training data, the goal is to find the probability distribution of the target task with insufficient training samples using the information acquired from the domain with sufficient data.

The target is to classify regions (zones) into text or non-text regions using the dataset collected from the newspapers mentioned before. Since the used dataset set is small, we can use transfer learning to enhance the accuracy and learning by using a dataset in another domain but with sufficient training samples. So, for the source domain $D_S$, the famous ImageNet [18] dataset is used to extract from its images the features of non-textual components as all of its images are colorful images of nature scenes, animals, etc. For textual feature extraction, APTI dataset [19] is used to capture textual features from it.

So, having a source domain $D_S$ with a $FS$ of $X_S$ captured from ImageNet and APTI datasets with labels $Y_S$ and a target domain $D_T$ with $FS$ captured from a few samples collected from the newspapers $X_T$, now the objective is to find the marginal probability distribution of $P(Y_T|X_T)$.

The whole training set for the source domain model $D_S$ consists of 100,000 samples, 50,000 64X64 images from ImageNet and another 50,000 64X64 (after resize) images from APTI. 80,000 samples used for training and the remaining 20,000 samples used for testing and validation. Mean has been subtracted from all training samples to have a zero mean as in Equation 1.

$$X = (X - \mu)/S$$  \hspace{1cm} (1)

Where $X$ is the target feature, $\mu$ is the feature’s mean and $S$ is the feature’s range of values $m-n$ (in this case $S=255$ which is the maximum intensity value of the pixel).

A D-CNN network is used to train a classifier to classify text and non-text regions/zones. D-CNN has proven its ability for image recognition tasks and have wide applications for video recognition, NLP and speech recognition. The architecture of the network used is as follows: Two convolutional layers of 64 feature maps of 3X3 kernel size each, followed by another convolutional layer of 128 feature maps of 2X2 kernel size and a dropout = 0.5 [20] used to control overfitting. Finally,
three consecutive fully connected layers of 1024, 512 and 512 nodes respectively followed by a softmax layer of 2 nodes for output classes (0 for text and 1 for non-text).

Adam optimization algorithm [21] is used to update network weights. The domain model $D_S$ achieved an accuracy = 99.14%. TL in this case was applied because we have a small and different dataset compared to the training set used in the source domain model $D_S$. After training the model on the source domain $D_S$, the model is re-trained on 10,000 training samples that were collected from the Arabic newspapers (documents were sliced into patches of size 64X64) for the target model $X_T$ but after freezing all the layers of the pre-trained model $D_S$. We froze all the layers to ensure that we do not unlearn the previously acquired knowledge from the pre-trained model. Results can be found at the end of this document (Fig. 8 and Fig. 9. Textual regions labeled in green and non-text regions labeled in red). As mentioned before, two techniques were used for classification, 1) zone based classification and 2) patch based classification. For zone/block based classification, ARLSA was used to segment the document image into zones/blocks then each block is sliced into small patches of size 64X64. These patches are presented to the network to classify each one of them as a text or non-text. After that, each zone/block is classified as text, if the number of the patches classified as text is greater than the half of the total number of patches of this zone as in Equation 2.

$$Zone_i = \begin{cases} 
    \text{text}, & \text{if } m > 0.5 \times n \\
    \text{non-text}, & \text{otherwise}
\end{cases}$$ (2)

Where $m$ is the number of patches classified as text for each zone, $n$ is the total number of patches, $i$ is the zone index because the document may have more than one zone and the parameter 0.5 was estimated by trial and error. For this technique, evaluation is based on number of zones classified as text or non-text.

For patches based classification, the same was done as the first technique by applying ARLSA first. Then each segmented zone is sliced into small patches of size 64X64. The difference is that these patches are fed to the CNN only without applying any rules as the previous techniques. The evaluation of this technique is relied on the number of patches classified in the whole 40 document images’ set.

3) Text lines and words segmentation: After labeling the document into text and non-text zones, text zones are segmented into lines and words. To segment lines, Vertical Projection Profile analysis ($VPP$) was used. After text lines segmentation, each text line is segmented into words using Horizontal Projection Profile analysis ($HPP$) as in Equation 3. $VPP$ and $HPP$ was used because the documents’ images that we work on are clean from noise.

$$VPP(y) = \sum_{0<x<w} f(x,y); HPP(x) = \sum_{0<y<h} f(x,y)$$ (3)

Where $f(x,y)$ is the input image, $w$ is the image width, $h$ is the image height, $VPP(y)$ is the sum of each row of the image and $HPP(x)$ is the sum of each column of the image. Fig. 4 shows the $VPP$ histogram. Lines can be segmented at the lowest histogram responses (where response is equal to zero).

![Figure 4: Vertical Projection Profile Histogram.](image1)

![Figure 5: Horizontal Projection Profile Histogram.](image2)

Fig. 5 shows the $HPP$ histogram. Words can be segmented at the lowest histogram responses (when response is equal to zero).

IV. RESULTS AND DISCUSSION

A. Datasets

As mentioned before, three datasets were used. ImageNet [18], APTI [19] and a dataset collected from Arabic printed newspapers.

1) Arabic Printed Text Image database : it is a dataset for Arabic printed words. It contains 113,284 word images of several font types, styles and sizes. The database is used for competitions related to research of Arabic OCR. The database contains ten different font types with four font styles (Italic, Bold, Plain and Bold Italic) and with ten font sizes. APTI fonts are shown in Fig. 6.
2) ImageNet: is a large visual dataset prepared for object detection and recognition. It contains more than 14,000,000 images with 1000 different classes. The dataset has been used for years for their competitions and it became a very common dataset used for the tasks of image classification. The dataset was used for the purpose of extraction of the visual features of the images regardless the category of the image itself. So, as the target is to classify and differentiate between text and non-text components in the newspaper image, ImageNet was used to train the D-CNN the structure of the images with natural scenes, animals, etc. (non-text components).

3) Arabic Newspapers: A dataset of Arabic newspapers was collected from three different Arabic newspapers’ websites (Al-Sharq [11], Riyadh [12] and Al-Youm [13]). The collected newspapers’ images were extracted from its PDF format. All pages of the collected PDFs were converted into images to use them for training and testing in the proposed system. Dataset samples shown in Fig. 7.

Table I: Confusion matrix for zone based classification

<table>
<thead>
<tr>
<th>Actual values</th>
<th>Text</th>
<th>Non-text</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>188</td>
<td>32</td>
<td>220</td>
</tr>
<tr>
<td>Non-text</td>
<td>98</td>
<td>365</td>
<td>463</td>
</tr>
<tr>
<td>Total</td>
<td>286</td>
<td>397</td>
<td>683</td>
</tr>
</tbody>
</table>

The achieved results for zone based classification: 
Precision = 0.66, Recall = 0.855 and F-score = 0.745.

For patches based classification, total number of patches classified in 40 documents = 8499; 5650 zones are actual text patches and 2849 zones are actual non-text patches.

Table II: Confusion Matrix for Patches Based Classification

<table>
<thead>
<tr>
<th>Prediction outcome</th>
<th>Text</th>
<th>Non-text</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>4896</td>
<td>754</td>
<td>5650</td>
</tr>
<tr>
<td>Actual values</td>
<td>219</td>
<td>2630</td>
<td>2849</td>
</tr>
<tr>
<td>Total</td>
<td>5115</td>
<td>3384</td>
<td></td>
</tr>
</tbody>
</table>

B. Results

Training phase was done on a machine with AMD FX-8350 processor, 16 GB memory and NVidia GTX 1060 used for training the D-CNN. The training time of the D-CNN took around 28 continues hours. The system has been evaluated on 40 document images (10 skewed and 30 non-skewed) collected from Arabic newspapers. The evaluation was measured based on two techniques as mentioned earlier: using 1) zone/block based classification and 2) patches based classification. Results for both techniques in Table I and Table II. For zone based classification, total number of zones classified in 40 documents = 683; 220 zones are actual text zones and 463 zones are actual non-text zones. For zone based classification, total number of zones classified in 40 documents = 8499; 5650 zones are actual text zones and 2849 zones are actual non-text zones.
The achieved results for patches based classification: 
Precision = 0.96, Recall = 0.87 and F-score = 0.91.

V. CONCLUSION

We introduced a method for Arabic document layout analysis using deep learning. The main motivation of this research is to develop an algorithm for analyzing Arabic newspapers using Deep Convolutional Neural Network (D-CNN) as it achieves great results for tasks related to machine vision. We showed that using a proper pre-trained model and a small dataset we could achieve good results by applying the concept of transfer learning.

We used two classification methods: 1) zone based classification and 2) patches based classification. The system has been evaluated on 40 document images (10 skewed and 30 non-skewed); the best results achieved for text and non-text classification are Precision = 0.96, Recall = 0.87 and F-score = 0.91. Using ImageNet and APTI datasets, we trained a network to discriminate images and text, and then we fine-tuned the network using a very small dataset collected from Arabic popular newspapers’ images. ImageNet was used to extract the visual features of the images regardless of their categories because we only need to discriminate between text and non-text components; APTI dataset was used as well to capture the visual features of the text.

REFERENCES


Figure 8: (a), (d): original document images. (b), (e): zone based classification. (c), (f): patches based classification (green: text, red: non-text).
Figure 9: (a), (d): original document images. (b), (e): zone based classification. (c), (f): patches based classification (green: text, red: non-text).