Layout Analysis for Arabic Historical Document Images Using Machine Learning

Syed Saqib Bukhari, Thomas M. Breuel
Technical University of Kaiserslautern, Germany
bukhari@informatik.uni-kl.de, tmb@informatik.uni-kl.de

Abedelkadir Asi, Jihad El-Sana
Ben-Gurion University of the Negev, Israel
abedas@cs.bgu.ac.il, el-sana@cs.bgu.ac.il

Abstract

Page layout analysis is a fundamental step of any document image understanding system. We introduce an approach that segments text appearing in page margins (a.k.a. side-notes text) from manuscripts with complex layout format. Simple and discriminative features are extracted in a connected-component level and subsequently robust feature vectors are generated. Multilayer perception classifier is exploited to classify connected components to the relevant class of text. A voting scheme is then applied to refine the resulting segmentation and produce the final classification. In contrast to state-of-the-art segmentation approaches, this method is independent of block segmentation, as well as pixel level analysis. The proposed method has been trained and tested on a dataset that contains a variety of complex side-notes layout formats, achieving a segmentation accuracy of about 95%.

1 Introduction

Manually copying a manuscript was the ultimate way to spread knowledge before printing houses were established. Scholars added their own notes on page margins mainly because paper was an expensive material. Historians regard the importance of the notes’ content and the role of their layout; these notes became an important reference by themselves. Hence, analyzing this content became an inevitable step toward a reliable manuscript authentication [11] which would subsequently shed light on the manuscript temporal and geographical origin.

Figure 1. Arabic historical document image with complex layout formatting due to side-notes text.

Physical structure of handwritten historical manuscripts imposes a variety of challenges for any page layout analysis system. Due to looser formatting rules, non-rectangular layout and irregularities in location of layout entities [2, 11], layout analysis of handwritten ancient documents became a challenging research problem. In contrast to algorithms which cope with modern machine-printed documents or historical documents from the hand-press period, algorithms for handwritten ancient documents are required to cope with the above challenges.

Page layout analysis is a fundamental step of any document image understanding system. The analysis process consists of two main steps, page decomposition and block classification. Page decomposition seg-
ments a document image into homogeneous regions, and the classification step attempts to distinguish among the segmented regions whether they are text, picture or drawing. Later on, the text regions are fed into a recognition system such as, Optical Character Recognition (OCR), to retrieve the actual letters and words which correlate to the characters in the manuscript.

In this paper, we introduce an approach that segments side-notes text from manuscripts with complex layout formatting (see Figure 1). It extracts and generates feature vectors in a connected-component level. Multi-layer perception classifier, which has been already used for page-layout analysis by Jain and Zhong [9], was exploited to classify connected components to the relevant classes of text. A voting step is then applied to refine the resulting segmentation and produce the final classification. The suggested approach is independent of block segmentation, as well as pixel level analysis.

In the rest of the paper, we overview previous work, present our approach in detail, report experimental results, and finally we conclude and suggest directions for future work.

2 Related Work

Due to the challenges in handwritten historical documents [2], applying traditional page layout analysis methods, which usually address machine-printed documents, is not applicable. Methods for page layout analysis can be roughly categorized into three major classes: bottom-up, top-down and hybrid methods [12, 15, 7]. In top-down methods, the document image is divided into regions which are classified and refined according to pre-defined criteria. Bottom-up approaches group basic image elements, such as pixels and connected components, to create larger homogeneous regions. Hybrid schemes exploit the advantages of top-down and bottom-up approaches to yield better results.

Recently, Graz et al. [8] introduced a binarization-free approach which employs the Scale Invariant Feature Transform (SIFT) to analyze the layout of handwritten ancient documents. The proposed method suggests a part-based detection of layout entities locally, using a multi-stage algorithm for the localization of the entities based on interest points. Support Vector Machine (SVM) was used to discriminate the considered classes. Kise et al. [10] introduced a page segmentation method for non-Manhattan layout documents. Their method is based on connected components analysis and exploits the Area Voronoi Digarams to segment the page. Bukhari et al. [5] presented a segmentation algorithm for printed document images into text and not-text regions. They examined the document in the level of connected components and introduced a self-tunable training model (AutoMLP) for distinguishing between text and non-text components. Connected components shape and context were utilized to generate feature vectors. Moll et al. [14] suggested an algorithm that classifies individual pixels. The approach is applied on handwritten, machine-printed and photographed document images. Pixel-based classification approaches are time-consuming in comparison to block-based and component-based approaches.

Page layout analysis was also posed as a texture segmentation problem in literature. For texture-based approaches see reviews in [13, 16]. Jain and Zhong [9] suggested a texture-based language-free algorithm for machine-printed document images. A neural network was employed to train a set of masks which were designated to be robust and distinctive. Texture features were obtained by convolving the trained masks with the input image. Shape and textural image properties motivated the work introduced by Bloomberg in [3]. In this work, standard and generalized (multi-resolution) morphological operations were used. Later on, Bukhari et al. [6] generalized Bloomberg’s text/image segmentation algorithm for separating text and non-text components including halftones, drawings, graphs, maps, etc. The approach by Won [19] focuses on the combination of a block based algorithm and a pixel based algorithm to segment a document image into text and image regions.

Ouwayed et al. [17] suggested an approach to segment multi-oriented handwritten documents into text lines. Their method addressed documents with complex layout structure. They subdivided the image into rectangular cells, and estimated text orientation in each cell using projection profile. Then, cells are merged together into larger zones with respect to their orientation. Wigner-Ville Distribution was exploited to estimate the orientation within large zones. This method could not yield accurate segmentation results due to some assumptions that were adopted by the authors. When a window contains several writings in different orientations, the authors assumed that the border between the two types of writing could be detected by finding the minimum index in the projection profile to refine the cells subdivision. However, this border is not always obvious and detecting the minimum index from the projection profile becomes a real challenge when side-notes are written in a flexible writing style (see Figure 1). One can also notice that the robustness of this approach could be negatively affected once side-notes text have the same orientation as main-body text and the two types of text have no salient space between them. In
this case the method would not distinguish between the
two coinciding regions and erroneous text-lines would
be extracted.

3 Method

Conventional methods for geometric layout analy-
sis could be an adequate choice to tackle the side-notes
segmentation problem when main-body and side-note
text have salient and differentiable geometric proper-
ties, such as: text orientation, text size, white space lo-
cations, etc. However, layout rules have not necessarily
guided the scribes of ancient manuscripts, as a result,
complex document images became common. These
documents contain non-uniform and/or similar geomet-
ric properties for both main-body and side-notes text; a
fact that makes the developing of a method which could
gracefully cope with this type of documents a challeng-
ing task.

Our approach utilizes machine learning technique to
meet the challenges of this problem. In general, clas-
sifier tuning is a hard problem with respect to the opti-
mization of their sensitive parameters, e.g., learning ‘C’
and gamma of SVM classifier.

Here, we are using MLP classifier for segmenting
side-notes from main-body text in complex Arabic doc-
uments. This approach is based on a previous work of
Bukhari et al. [5]. The main reason of using MLP clas-
sifier over others is that it achieves good classification
once it is adequately trained as well as being scalable.
However, a major difficulty of its use has been the re-
quirement for manual inspection in the training process.
They are hard to train because their performance is sen-
sitive to chosen parameter values, and optimal param-
eter values depends heavily on the considered dataset.
The parameters optimization problem of MLPs could
be solved by using grid search for classifier training.

But grid search is a slow process. Therefore in order
to overcome this problem we use AutoMLP [4], a self-
tuning classifier that can automatically adjust learning
parameters.

3.1 AutoMLP Classifier

AutoMLP combines ideas from genetic algorithms
and stochastic optimization. It trains a small number
of networks in parallel with different learning rates and
different numbers of hidden layers. After a small num-
ber of training cycles the error rate of each network is
determined with respect to a validation dataset accord-
ing to an internal validation process. Based on valida-
tion errors, the networks with bad performance are re-
placed by the modified copies of networks with good

3.2 Feature Extraction

As it widely known, once reliable features are ex-
tracted adequately, they could leverage the accuracy of
the classification step. Representative feature vectors
could be of high dimensions, however, in this work we
extract simple feature vectors, yet distinguishable and
representative ones. One can notice that the raw shape
of a connected component itself incorporates important
discriminative data - such as density - for classifying
main-body and side-notes text, as shown in Figure 2.
The neighborhood of a connected component plays also
a salient role towards a perfect classification. Figure 2
shows surrounding regions of main-body and side-notes
components. We refer to a connected component with
its predefined neighborhood as context.

We used the following features to generate discrimi-
native feature vectors:

- **Component Shape**: For shape feature genera-
tion, each connected component is downscaled to a
64 × 64 pixel window size if either width or height
of the component is greater than 64 pixels, other-
wise it is fit into the center of a 64 × 64 window.
This type of rescaling is used in order to exploit the
incorporated information in a components shape
with respect to its size.

We utilize additional four important characteristics
of connected components:

1. **Normalized height**: the height of a compo-

2. **Foreground area**: number of foreground pix-

3. **Relative distance**: the relative distance of a

4. **Orientation**: the orientation of a connected

We utilize additional four important characteristics
of connected components:

1. **Normalized height**: the height of a compo-

2. **Foreground area**: number of foreground pix-

3. **Relative distance**: the relative distance of a

4. **Orientation**: the orientation of a connected


will elaborate later (component context). The regions’ orientation is estimated based on directional projection profile for 12 angles with a step of 15° i.e. from $-75^\circ$ to $90^\circ$. The profile with robust alternations between peaks and valleys has been chosen. We compute a score $s$ for each rotation angle [18], then, the angle that corresponds to the profile with the highest score is chosen as the final orientation. The score is calculated according to Eq. 1.

$$s = \frac{1}{N} \sum_{i=0}^{N} \left( \frac{y^{(n)}_{h} - y^{(n)}_{l}}{h^{(n)}} \right)$$

where $N$ is the number of peaks found in the profile, $y^{(n)}_{h}$ is the value of the $n$th peak, and $y^{(n)}_{l}$ is the value of the highest valley around the $n$th peak. In our case $h^{(n)} = 1$ because our dataset does not contain non-rectangular document images; which was possible in [18].

Together with these four discrete values, the generated shape-based feature vector is of size $64 \times 64 + 4 = 4100$.

- **Component Context:** To generate context-based feature vector, each connected component with its surrounding context area is rescaled to a $64 \times 64$ window size, while the connected component is kept at the center of the window. The considered neighborhood is calculated adaptively as a function of component’s *width* and *height* (denoted by $w$ and $h$ respectively), and is $w_{\text{factor}} \times w$ by $h_{\text{factor}} \times h$, where $w_{\text{factor}}$ is always greater than $h_{\text{factor}}$ because of the horizontal nature of Arabic script. $w_{\text{factor}}$ and $h_{\text{factor}}$ were obtained experimentally and they equal 5 and 2, respectively. The rescaled main-body and side-notes components context are shown in Figure 2. The size of context-based feature vector is $64 \times 64 = 4096$. In this way, the size of a complete shape-based and context-based feature vector is $4100 + 4096 = 8196$.

### 3.3 Training dataset

Our dataset consists of 38 document images which were scanned at a private library located at the old city of Jerusalem and other samples which were collected from the Islamic manuscripts digitization project at Leipzig university library [1]. The dataset contains samples from 7 different books. From the 38 document images, 28 samples were selected as training set and the remaining 10 were used as testing set.

Main-body text and side-notes text are separated and extracted from the original document images to generate the ground truth for the training phase. The same process is applied on the testing set for evaluation purposes. Around 13 thousand main-body text components and 12 thousand side-notes components are used for training AutoMLP classifier. A segmented image generated by applying the trained MLP classifier is shown in Figure 3(a) and Figure 3(b). It is widely known that generalization is a critical issue when training a model, namely, generating a model that has the ability to predict reliably the suitable class of a given sample that does not appear in the training set. In our case, we are using a relatively small amount of document images for training which is still able to show the effectiveness of our approach.

In order to improve the segmentation results we use a post-processing step based on relaxation labeling approach which is described below.

### 3.4 Relaxation Labeling

We improve the segmentation results applying nearest neighbor analysis and using class probabilities for refining the class label of each connected component. For this purpose, a region of $150 \times 150$ is selected from the document by keeping the target connected component at the center. Several region sizes were tested and the one that yielded the highest segmentation accuracy (F-measure; discussed in next section) was chosen (as
Figure 3. (a) and (b) depict the segmentation of two samples before post-processing. (c) and (d) represent the final segmentation, respectively.

appears in Figure 4). The probabilities of connected components within the selected regions were already computed during the classification phase. The labels of connected components were updated using the average of main-body and side-notes component probabilities within a selected region. To illustrate the effectiveness of the relaxation labeling step, some segmented images are shown in Figure 3(c) and Figure 3(d).

4 Experimental Results

As stated above, our dataset contains 38 document images from which 10 images were chosen to build the testing set and it contains different images from different books. We test the performance of our approach using images with various writing styles and different layout structures which were not used for training.

Pixel-level ground truth has been generated by manually assigning text in the documents of the testing set with one of the two classes, main-body or side-notes text. Several methods to measure the segmentation accuracy have been reported in literature. We evaluate the segmentation accuracy by adopting the F-measure metric which combines precision and recall values into a single scalar representative. It guarantees that both values are high (conservative), in contrary to the average (tolerant) which does not hold this property. For example, when precision and recall both equals one, the average and F-measure will both be one, but, if the precision is one and the recall is zero, the average would be 0.5 and the F-measure would be zero. Therefore, this measure has been adopted as it reliably measures the segmentation accuracy. Precision and recall are estimated according to Eq. 2 and Eq. 3, respectively:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (2)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (3)
\]

where True-Positive(TP), False-Positive(FP) and False-Negative(FN) with respect to side-notes, are defined as following:

- **TP**: side-notes text classified as side-notes text.
- **FP**: side-notes text classified as main-body text.
- **FN**: main-body text classified as side-notes text.

Likewise, these metrics can also be defined with respect to main-body text. Once we have the precision and recall counts, F-measure is calculated according to Eq. 4.

\[
\text{F-Measure} = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Recall}) + \text{Precision}} \quad (4)
\]

Assigning \( \beta = 1 \) induces equal emphasis of precision and recall on F-measure estimation. F-measure for
<table>
<thead>
<tr>
<th>Window Size</th>
<th>Main-body F-Measure (%)</th>
<th>Side-notes F-Measure (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>91.37</td>
<td>90.74</td>
</tr>
<tr>
<td>100</td>
<td>94.34</td>
<td>93.93</td>
</tr>
<tr>
<td>150</td>
<td>95.02</td>
<td>94.68</td>
</tr>
<tr>
<td>200</td>
<td>94.65</td>
<td>94.22</td>
</tr>
<tr>
<td>250</td>
<td>93.91</td>
<td>93.35</td>
</tr>
</tbody>
</table>

Table 1. Performance evaluation of our method for both main-body and side-notes text with different post-processing window sizes.

both main-body and side-notes text with different post-processing window sizes is shown in Table 1. Note that the optimal window size is 150.

5 Discussion and future work

We have presented an approach for segmenting side-notes text in Arabic manuscripts with complex layout formats. Machine learning was exploited to classify connected components to the relevant class of text. We presented a set of simple and reliable features that yield almost perfect segmentation. A voting step was applied to refine the resulting segmentation and produce the final classification. For side-notes, a segmentation accuracy of about 95% was achieved. We think that a better model can be trained with a larger amount of samples thus it could be generalized and subsequently perfect segmentation would be achievable.

Our future work will focus on improving some aspects of the algorithm. Due to the fact that side-notes and main-body text were usually written by different writers, scribe writing style would definitely enhance the reliability of our feature vectors. Additional efforts will be invested in making the post-processing step as efficient as possible, and even avoiding it in some cases.

This research was supported in part by the Israel Science Foundation grant no. 1266/09, the German Research Foundation (DFG) under grant no. FI 1494/3-1 and the Lynn and William Frankel Center for Computer Science at Ben-Gurion University of the Negev.

References


