International Journal on Digital Libraries (2020) 21:329–342
https://doi.org/10.1007/s00799-020-00280-w

Historical document layout analysis using anisotropic diffusion and geometric features

Galal M. BinMakhashen[1] · **Sabri A. Mahmoud**[1]

Received: 12 November 2018 / Revised: 21 December 2019 / Accepted: 5 January 2020 / Published online: 23 January 2020

Abstract

There are several digital libraries worldwide which maintain valuable historical manuscripts. Usually, digital copies of these manuscripts are offered to researchers and readers in raster-image format. These images carry several document degradations that may hinder automatic information retrieval solutions such as manuscript indexing, categorization, retrieval by content, etc. In this paper, we propose a learning-free and hybrid document layout analysis for handwritten historical manuscripts. It has two main phases: page characterization and segmentation. First, the proposed method locates main-content initially using top-down whitespace analysis. It employs anisotropic diffusion filtering to find whitespaces. Then, it extracts template features representing manuscripts' authors writing behavior. After that, moving windows are used to scan the manuscript page and define main-content boundaries more precisely. We evaluated the proposed method on two datasets: One set is publicly available with 38 historical manuscript pages, and the other set of 51 historical manuscript pages that are collected from the online Harvard Library. Experiments on both datasets show promising results in terms of segmentation quality of main-content that reaches up to 98.5% success rate.

Keywords Image segmentation · Document indexing · Document retrieval · Document analysis · Whitespace analysis · Anisotropic diffusion filtering · Geometric feature

1 Introduction

Document layout analysis (DLA) is an essential research topic of document understanding systems for decades [22, 31]. DLA methods, at their early stages, were designed for simple document layouts. They were considered as subroutines of the preprocessing phase [44]. Eventually, researchers have considered complex and diverse document layouts to analyze. Consequently, DLA is recognized as a dedicated research field [29].

Complex layout analysis is not a trivial task. Such documents with such complex layouts usually have several issues include writing style, document structures and conditions. The text in these documents may be written by several writers using different fonts and sizes. Moreover, such documents may include arbitrary blocks of paragraphs, images, tables, etc. In addition, they may suffer from aging, faint-text, ink-bleeding, etc. So, all these issues make the complex layout analysis a challenging task.

Handwritten historical manuscripts are considered complex layouts [5]. They may not follow a unique writing style, font type, or size. Consequently, the writing maintains no regular spacing between words, text-lines, or paragraphs. Different writers may write comments on the same page (i.e., side-notes).

In general, DLA methods are categorized into two strategies: bottom-up or top-down. A bottom-up strategy starts by analyzing a document at some small element levels such as pixels. Then, it groups the homogeneous ones to form larger blocks such as words, text, or figures. This grouping is continued until reaching some predefined stop conditions. In contrast, a top-down analysis starts at large document regions and divides them into smaller homogeneous zones such as splitting a region into text and figure. Also, this strategy has some predefined stop conditions such as stop splitting at the level of text-lines, words, etc.

Both strategies have strengths and weaknesses. On the one hand, a bottom-up strategy is normally slower to analyze

✉ Galal M. BinMakhashen
binmakhashen@kfupm.edu.sa

[1] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Table 1 Strengths and weaknesses of DLA strategies

Metrics	Bottom-up	Top-down
Performance		
Speed	Slow-moderate	Moderate-high
Accuracy	Moderate-high	Low-moderate
Parameterized	Often	Rare
Layouts		
Complex	Yes	NA
Regular	Yes	Yes

documents in comparison with the top-down strategy [39]. However, a top-down strategy is less accurate at determining zones' boundaries precisely [39]. Table 1 lists some crucial factors of top-down and bottom-up strategies.

In addition, another strategy combines both mentioned strategies to analyze documents. This type of DLA is known as a hybrid strategy such as [34,42]. The hybrid strategy starts the analysis by either top-down method then a bottom-up or the opposite way, but in both cases, it is carefully designed to avoid analysis of the weakness of both core strategies.

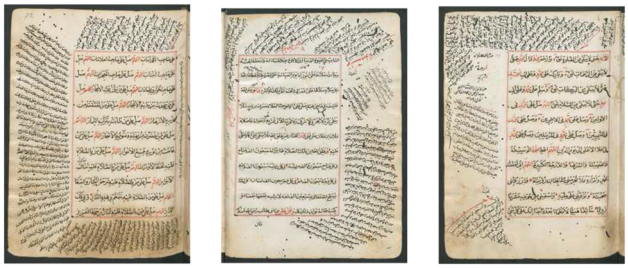
In this paper, we propose a hybrid learning-free DLA for Arabic historical manuscripts. The manuscripts are handwritten and mostly contain text. Moreover, they may be written by multiple writers in unconstrained style. Some examples of these documents are shown in Fig. 1.

Our proposed method has two main outcomes. First, it identifies historical manuscripts using main-content instead of traditional document titles. This is useful for digital libraries because some historical manuscripts were incompletely collected. So, title pages may be missing or titles are severely degraded and tough to be transcribed. Consequently, digital libraries can retrieve historical manuscripts-based main-content. Second, it proposes a whitespace analysis method for handwritten documents that employs anisotropic diffusion filtering (ADF). To our knowledge, the whitespace analysis method has not been used to analyze the handwritten document [32].

The rest of the paper is organized as follows. Section 2 discusses the related work. A detailed description of the proposed method is presented in Sect. 3. Experiments setup and results are given in Sect. 4. Final remarks and future work are given in Sect. 5.

2 Related work

Document layout analysis was approached by either learning-based [10] or learning-free [4] methods. The learning-based methods are divided into two categories with respect to the



(a) Islamic Heritage Project (IHP), Harvard library

(b) Bukhari et al. [10] dataset

Fig. 1 Samples of Arabic historical manuscripts

input data: pixel-based as in [6,14], or feature-based as in [10,46].

Pixel-based methods generate machine learning models using pure pixel intensities. For instance, Baechler et al. [5] proposed two-stage semi-automatic method for historical documents using a multi-layer perceptron (MLP). The authors suggested human interaction to determine document final segmentation.

Usually, standard training of MLP model requires tuning its hyperparameters. These hyperparameters are data-dependent; they are found empirically. Consequently, to build the best MLP model, several training sessions should be conducted. Due to these issues, automatic MLP (auto-MLP) was used to find model hyperparameters automatically [10].

In feature-based methods, document structures' attributes are used. Document structures' features can be categorized into global or local features. Usually, global features reduce the amount of input data required to train machine learning models. For instance, Chen et al. in [12] suggested using superpixels intensities instead of pure pixel values in model training. Another study by Le et al. [25] extracted geometric features from document components and used them to build machine learning model.

However, local features may produce a larger number of input data than global features. For example, Scale Invariant Feature Transform (SIFT) approach represents each salient image point by 128 feature points as in [20,36]. Another example is a Gabor filter that produces several features that depend on the number of scales and directions used in the filtering [4,15,30].

Recently, deep learning methods are used to auto-generate features [14]. Unlike traditional feature extraction methods, the auto-generated features are difficult to be described because they are generated out of some random processes that lead to deep learning model convergence [23]. Convolution auto-encoder (CAE) is one example method that can generate features for text-line extraction [47]. Moreover, several studies have considered auto-generated features for document layout analysis such as [13,37,38].

There are other machine learning algorithms that have been used in DLA such as Support Vector Machines (SVM) [17,19,20,45] and Gaussian Model Mixtures (GMM) [18,45,46].

In general, learning-based methods depend heavily on the robustness of features . Recently, Mehri et al. [29] compared nine texture feature extraction methods for DLA. The study recommended Gabor feature extraction for text versus non-text classification. They found that Gabor features can capture geometric writing features better than the tested methods in their study. However, they were neither considered handwritten documents nor Arabic documents in their method evaluation [29].

Unlike learning-based, learning-free methods may perform layout analysis faster with satisfactory performance such as [43]. Unfortunately, there are few learning-free DLA methods for historical manuscripts that can be found in the literature. To mention some examples, Bulacu et al. [11] suggested contour tracing approach to extract handwritten text from manuscripts of the Dutch Queen Archive. Asi et al. [4] extracted manuscripts main-content using Gabor filtering and energy minimization graph-cut algorithm. The main limitation of Asi's method is assuming the existence of multiple writers of each page. So, they did not evaluate their method using historical documents written by a single writer.

In this study, we extend the work of Asi et al. [4] by waiving the multi-writer constraint. This study extracts the main-content from historical manuscripts that have multiple text blocks on each page. Unlike Asi et al. [4] method, the proposed method conducts DLA using a hybrid strategy. First, a novel top-down approach is developed using anisotropic diffusion to find separating whitespace on handwritten documents. Then, a set of geometric features is extracted to characterize the main-content writing style. Second, a bottom-up analysis strategy is applied using moving-window and connected-component analysis to generate pages' main-content boundary map.

3 Proposed DLA method

Our method is learning-free hybrid analysis approach that treats each manuscript page as a unique example. The method combines bottom-up and top-down analyzes to extract main-content and side-notes from Arabic historical manuscripts. It starts by preprocessing an input manuscript page. The preprocessing consists of two steps: binarization, and noise removal. In binarization, it performs adaptive binarization using Sauvola et al. [35] that reduces the effect of unbalanced illumination and shadows. Then, it removes image noise as described in Sect. 3.1.2. Pages' main-content is initially located using whitespace masks that are built by using anisotropic diffusion and whitespace scanning methods. After that, each manuscript page-image is characterized by extracting geometric template features. Further details on main-content initial localization and characterization are given in Sect. 3.2. Second, a bottom-up analysis is conducted by moving three windows in four directions: left, right, top and down, to extract features and match them with template ones. The moving windows are stopped based on some pre-determined conditions as described in Sect. 3.3. Finally, K-means clustering is conducted to define a separating path between main-content and other regions. Figure 2 illustrates a general overview of the proposed method.

3.1 Preprocessing

3.1.1 Binarization

Although binarization may yield additional noise in a resultant binary image, it reduces processing time on subsequent phases because of data reduction. In binary images, the analysis considers only one image channel in comparison with RGB channels. In addition, only two values [0 or 1] are considered in binary images rather than 256 grayscale intensities in other formats.

Binarization methods can be divided into global or local approaches. A method is called global binarization if it computes a single binarization threshold that is used to classify image pixels to either foreground or background. Otsu approach [33] is a well-known global binarization method. However, local binarization methods use multiple thresholds to binarize an image [35,40]. Figure 3b shows a global binarization result using the Otsu method. Figure 3d shows a result of a local binarization method by Sauvola and Pietikainen [35]. As can be noticed, the amount of binarization leftover (i.e., noise) is larger in Fig. 3b in comparison with Fig. 3d. However, not all local binarization is suitable for historical manuscripts. For instance, in Fig. 3c, local binarization affected text structures and removed parts of them. This effect is due to using multiple thresholds from non-overlapping image-block and without normalization, which led to a severe removal of text content. Sauvola and Pietikainen method computes dynamic thresholds by normalizing them using a dynamic range computed overall image-blocks. In this work, we adopted the Sauvola and

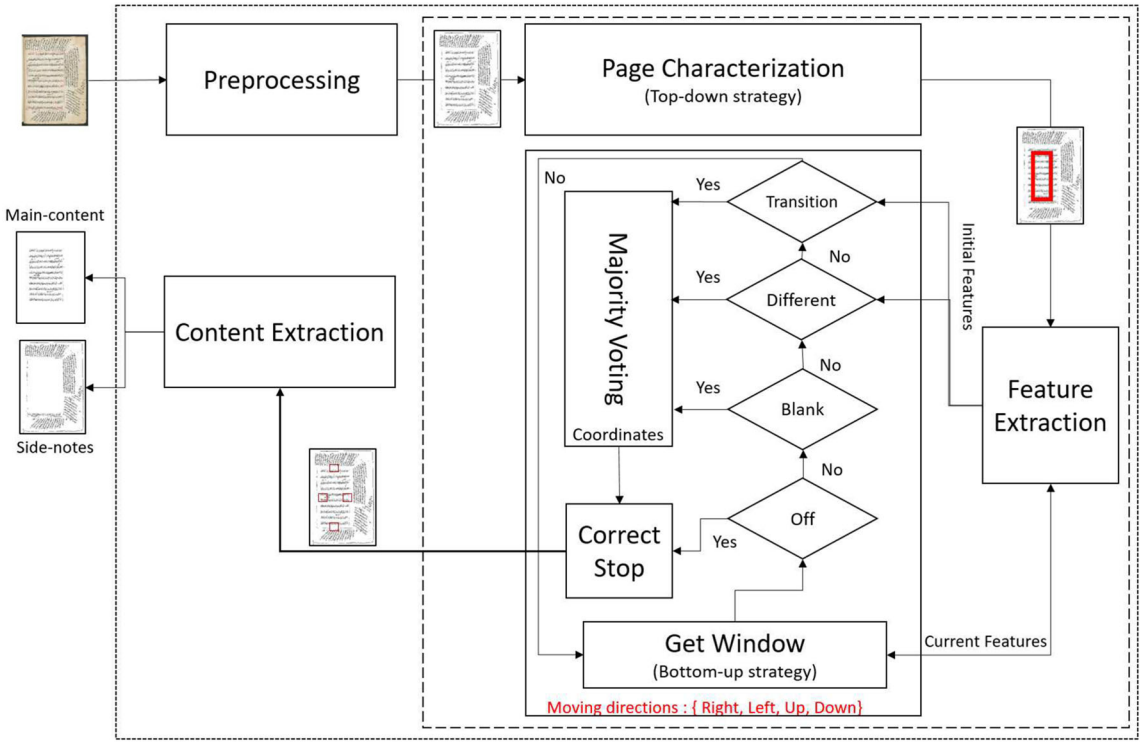
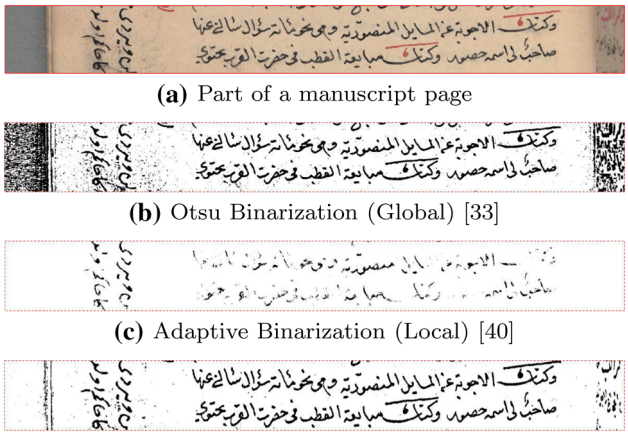


Fig. 2 A general overview of the proposed DLA method



(a) Part of a manuscript page

(b) Otsu Binarization (Global) [33]

(c) Adaptive Binarization (Local) [40]

(d) Sauvola & Pietikainen Binarization (Local) [35]

Fig. 3 Global and local binarization comparison

Pietikainen binarization method to binarize historical documents.

3.1.2 Noise removal

The main aim of the noise removal step is to ensure that subsequent processes will consider significant connected-components for the analysis. Therefore, small elements such as diacritics, dots, commas, and binarization's leftover artifacts are considered as noise.

We implemented a simple noise removal procedure based on connected-component (CC) characteristics. Let Ca be the connected-component area and let C_{avg} be the connected-components' average area computed as:

$$C_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} Ca_i \tag{1}$$

where n is the number of significant CCs in a manuscript page. The cleaned image is computed by removing all relatively small CCs, which are less than C_{avg} as:

$$I_c = \begin{cases} Ca_j \in I_c & \text{if } Ca_j \geq C_{\text{avg}} \quad \text{where} \quad j = 1, 2, \ldots, n \\ Ca_j \notin I_c & \text{Otherwise} \end{cases} \tag{2}$$

where I_c is the filtered manuscript image.

3.2 Page characterization

Manuscript characterization involves initial localization of main-content to compute content's template features. First, the page's main-content localization is realized using whitespace analysis based on ADF and Static Whitespace Location (SWL). The localization generates two masks representing main-content coordinates, ADF and SWL masks. Then, these

masks are integrated to define the main-content's coordinates. Second, a set of template features that captures the manuscript's author writing characteristics is extracted.

3.2.1 Adaptive main-content localization

Adaptive localization of the page's main-content is an initial step in estimating the writing characteristics of the manuscript. Moreover, we treat every page of a manuscript as a unique example because main-content writings of the same manuscript may be written by different writers too [10].

A. *Whitespace localization using anisotropic diffusion*
Usually, whitespace analysis is used to analyze printed documents because they preserve structured layouts such as [8,9,24,43]. In these examples, maximal whitespace rectangles are located to segmented different document blocks such as text-columns, figures, etc.
In this work, whitespace analysis is used to analyze handwritten historical manuscripts. Instead of whitespace rectangles, we use major whitespace detection that could be non-rectangle shaped, but it represents a gap between two separate regions. The major whitespace should be either long horizontally or vertically. By long, we mean the major whitespace spans through several text-lines or words vertically or horizontally, respectively.
Empirically, we found that a whitespace with a width or height of at least one-third a page's width or height is a good candidate for major whitespace. Usually, major whitespaces are located at the transition zones such as between text-lines, or different blocks.
Locating major whitespace is a challenging task because historical manuscripts are mostly text with dense and unconstrained writing styles. To resolve this issue, an anisotropic diffusion filtering (ADF) is applied to the manuscript images to emphasize foreground/background separation and to boost the segmentation process. This is computed using the second derivative of filtered images that produces high responses representing candidate whitespace. The standard form of the oriented anisotropic Gaussian filter impulse response is given in Eq. (3).

$$g^{\theta}_{(u,v,\sigma_u,\sigma_v,\theta)} = \frac{1}{\sqrt{2\pi}\sigma_u} e^{\left(-\frac{u^2}{2\sigma_u^2}\right)} * \frac{1}{\sqrt{2\pi}\sigma_v} e^{\left(-\frac{u^2}{2\sigma_v^2}\right)} \quad (3)$$

where $*$ is a convolution operation, σ_u and σ_v are the standard deviations of both frequency components u and v which represent both directions of the angle θ and the orthogonal to θ, respectively. Both frequencies are defined in Eq. (4):

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4)$$

Interested readers can find further details about the implementation of ADF in [21].
As noticed in Eq. (3), it filters foreground and background regions by steering the filter locally against text-strokes in orthogonal directions. In this sense, the ADF scale parameters can be computed based on writing characteristics such as text-strokes or gaps. Therefore, we calculate the ADF scales using text geometric attributes. Hence, the σ_u and σ_v are computed as follows:

$$\sigma_u = \alpha \times \mu_{Hcc} + \beta \times \mu_{Lgaps} \quad (5)$$

where μ_{Hcc} is the average height of all dominant connected-components, μ_{Lgaps} is the average height of the vertical gaps between connected-components, and α, β are two weights values set manually to reduce the variability effect of connected-components heights and vertical gaps.

$$\sigma_v = \alpha \times \mu_{Wcc} + \beta \times \mu_{PAWgaps} \quad (6)$$

where μ_{Wcc} and $\mu_{PAWgaps}$ are the average width of dominant connected-components and the horizontal gaps between them, respectively.
To ensure ADF avoids the effect of writing fluctuation, the ADF is applied on a range of angles $[10° \pm \frac{\pi}{2}]$ and $[10° \pm (\pi \times 2)]$ in the vertical and horizontal directions, respectively.
In historical Arabic manuscripts, written text-notes appear on pages' margin space around the main-content (i.e., side-notes). In the case of densely written notes, main-content and side-notes may not be clearly separable because some text components of side-notes or main-content may touch the other type of text. So, clear separation may be disturbed. Figure 1 shows the manuscript pages with different levels of text density. The first example in Fig. 1b represents a situation where main-content components touch side-note components. In similar situations, ADF may not perfectly help in separating main-content from side-notes. However, an estimate of separating path can be delivered based on some whitespace clues found in other directions. For example, if the left-side of the main-content cannot be separated directly, then, top and down separating boundaries can be used to find an estimate of the left-side boundary.
Since the left and right boundaries of main-content are difficult to detect, we discuss ADF vertical response and scale parameter σ_u with illustrative examples shown in Figs. 4 and 5.
In Fig. 4, white components represent text components, and magenta components represent ADF high response. Figure 4 column (a) shows a long white component representing text-touching of text-lines. On the one hand, a

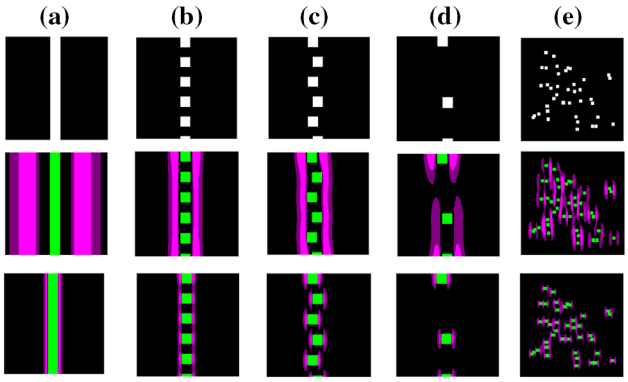


Fig. 4 Simulated layout examples. First row: input, second row: estimated σ_u, third row: fixed $\sigma_u = 15$; **a** vertically long component due to text-line touching, **b** regular spaced and aligned components, **c** regular spaced unaligned components, **d** large gaps and no-alignment between components, **e** randomly scattered similar components

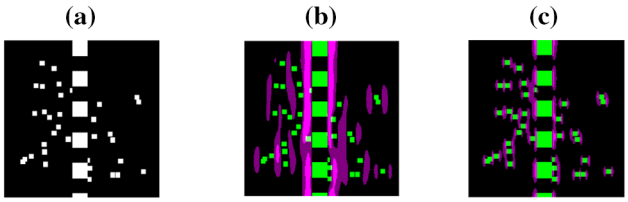


Fig. 5 Example of main component in middle of the image while some comments appear around, **a** input, **b** response with $S_x = \frac{\text{Avge}(CC_{\text{Height}})}{2}$, **c** fixed scale $S_x = 15$

max response of ADF using estimated σ_u from Eq. (5) is shown in Fig. 4a second row. The ADF response, in this case, leaves wide black gaps between the white component and the ADF response. These black gaps are safe zones that ADF high response (i.e., major-whitespace) leaves due to the filter's scale parameter. We use a safe zone for region segmentation. The safe zone width is affected by the ADF scale σ_u parameter. In Fig. 4a second row, the safe zone is wide because the large estimated scale was used $\sigma_u \approx 153$ pixels.

The safe zone is important to define the main-content boundary that does not interfere with other regions. The desired safe zone should be small enough between ADF response and the component. To investigate the effect of the ADF scale parameter, we set a fixed value for the scale as $\sigma_u = 15$ pixels in the second experiment. In this experiment, safe zones appear very close to a white component, and may lose their vertical continuity as shown in Fig. 4c third row.

The behavior of the ADF is further investigated for other situations. It is applied to other illustrative situations where vertical gaps are getting larger as shown in Fig. 4b, as well as white components lose their alignment as in Fig. 4c and d. The results of these illustrative experiments show that the ADF high responses can be preserved along the perpendicular direction of the strong gradient across foreground and background edges. Moreover, the scale value σ_u should be estimated based on text components. This allows the method to be more adaptive (see Fig. 4b–d second row). Hence, ADF maximum responses can be utilized to indicate whitespace locations between different regions in the handwritten text.

In Fig. 4e, white components are similar to each other and small. In this situation, ADF responded with short discontinuous coefficients. These short ADF responses are not qualified as major whitespace separation. Therefore, we can easily conclude that the page has one type of text. To explain this phenomena further, larger white components are added to the same illustrative example. These added components represent a different type of text. In this case, ADF responded with qualified major whitespaces that can be used to separate the two types of text (see Fig. 5b and c).

B. *Static whitespace localization*

Even though ADF has adaptively competed, it may fail to find main-content boundary precisely due to noise that affects scale estimation, or the existence of one type of text as in Fig. 4e. The failure of ADF in the previous step can be detected if the ADF returned weak responses. This can be formally stated as in Eq. (7):

$$T = \sum_{i=1}^{n} \text{ADF}_{H_i}(I) \tag{7}$$

where T is the ADF high response ADF_{H_i} accumulation. So, if $T \approx 0$, it indicates ADF failure.

Static Whitespace Location (SWL) detects whitespace gaps by scanning a manuscript page vertically and horizontally. It marks gaps that are greater than the average horizontal gap ω_s as whitespace. This threshold is not fixed for all pages, and it is computed in the preprocessing phase. The aim of SWL scans is to generate a whitespace mask that compliments ADF mask.

C. *ADF and SWL combination*

The resultant masks of both ADF and SWL are combined to generate better localization of the main-content initial region. The combination process is considered only if the ADF fails to detect the main-content according to Eq. (7). Investigating the strengths of both techniques, we found that ADF approach is robust to detect horizontal whitespace, while SWL is better to detect vertical whitespaces. Therefore, the generate masks are morphologically processed to remove short responses from the horizontal mask (ADF), and find the biggest middle gap in the vertical mask (SWL). Finally, the horizontal mask is added to the inverted vertical mask to generate the combined mask.

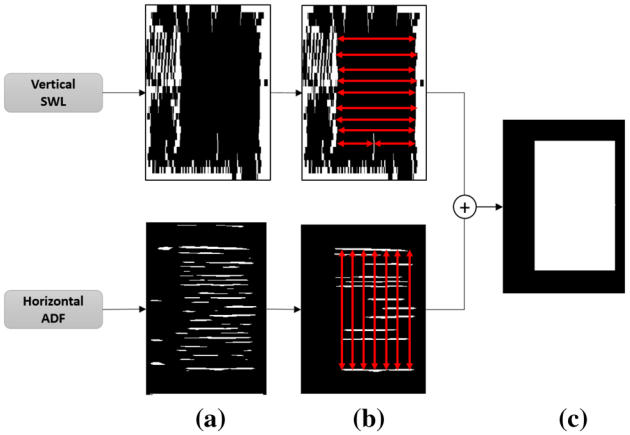


Fig. 6 ADF and SWL masks combination

Figure 6 shows the process of ADF and SWL masks combination.

3.2.2 Feature extraction

Once the initial main-content region is detected, a set of essential handwriting characteristics is extracted. The writing characteristics are mainly geometric measurements. The algorithm randomly selects eight frames over the initial main-content region. Frame w_f has a square shape, and its size is computed as follows:

$$w_f = (\alpha \times W_p) + (\beta \times G_l) \tag{8}$$

where α and β are two free parameters set to three and four, respectively, to create a frame of three times the average connected-components with W_p and four times the average vertical gaps G_l.

Eight geometric features are extracted from each frame as follows:

- Average height: This feature measures vertical stroke of main-content text. In comparison with main-content, the vertical strokes of side-notes are smaller due to the constrained writing-space. Therefore, connected-components' heights are considered to distinguish both types of writing styles.
- Average area: As in some cases, side-notes are written vertically. Therefore, the height feature alone may not be enough to distinguish main-content elements from side-notes. The text components' width is as important as the height. Consequently, the average area of text connected-components is used as a feature point.
- Foreground to background ratio (FBR): It represents how foreground pixels are distributed against background pixels of the main-content. In other words, it captures how big or small are handwritten elements in the main-content in comparison with side-notes. In addition, larger values of FBR represent either dense text or thicker writing strokes.
- Pixel density (PD): To support FBR, PD returns many foreground pixels that represent how dense is the main-content frame in terms of written elements.
- Average horizontal gaps (AHG): Although historical manuscripts are handwritten, still a writer can maintain reasonable spacing between his written words (i.e., writing habit characteristics). In addition, this characteristic may differ for the same writer in constraint writing. That is, if the same writer wrote both the main-content and side-notes, the horizontal spacing among words must be different due to space constraints.
- Average text line gap (ATG): The logic behind this feature is similar to AHG, but by considering the vertical direction of written text. Therefore, the average vertical whitespaces between text-lines are considered as a feature point.
- Distance transform (DT): Each binary foreground pixel (BFP) can be converted to a gray value that represents a distance between current BFP_i to its nearest BFP_j, where $i \neq j$ [28]. Figure 7 shows DT representation of the main-content sample frame in comparison with DT representation of the side-notes frame.
- Text orientation: Main-content text orientation is an important handwritten feature. There are different methods that can compute text orientation such as project profiles [7], Hough transform [41]. In this work, we developed a local text orientation estimation. The method starts by finding a centroid of each connected-component. Then, it determines the right neighbor of each of them. After that, local angles between these connected-components are computed. Finally, the average value of these computed angles is used as text orientation. To estimate text orientation, let v_{nm} be a vector between two neighboring connected-components n and m, and v_h is a reference horizontal vector on the x-axis. So, the component angle is computed as follows:

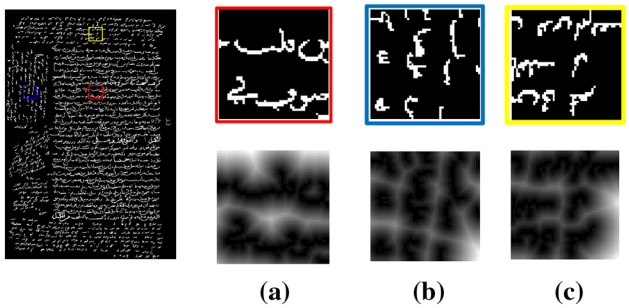


Fig. 7 Example of DT on **a** main-content frame; **b** side-notes frame (vertical text); **c** side-notes frame (flipped text)

$$\theta = \cos^{-1}\left(\frac{v_h \cdot v_{nm}}{\|v_h\| \times \|v_{nm}\|}\right) \quad (9)$$

where $\cdot$ is the dot product of two vectors and $\|.\|$ is the magnitude of the vectors.

3.3 Moving window analysis

3.3.1 Main-content boundary detection

The bottom-up analysis starts by moving three windows in four directions. Each window has a size of K that is determined by Eq. 8. Although initial main-content localization is performed faster due to image filtering and quick scanning of SWL, it lacks precision. Therefore, its main goal is to extract main-content template features that are used in a window-based analysis.

Three windows centered at the initial main-content region are moved toward left, right, up, and down directions of a manuscript page. The algorithm stops a moving window in one direction if at least two windows have met stop conditions (i.e., Majority-Voting).

There are four conditions to stop a moving-window analysis, namely blank-zone, off-boundary, transition-zone, or different-zone. Figure 8 shows an example of each stopping condition. A blank-zone and off-boundary are intuitive stop conditions. Once a moving window steps on a large empty background or reaches the end-border of a manuscript page, it must stop. Furthermore, the algorithm extracts some features as the windows moving. These features are matched with template features which were extracted previously to compute a score CS_i. If the score $CS_i > th_1$, then, the current moving window i is marked as a stop window because transition-zone condition is met. Similarly, for the different-zone condition, it uses another pre-computed

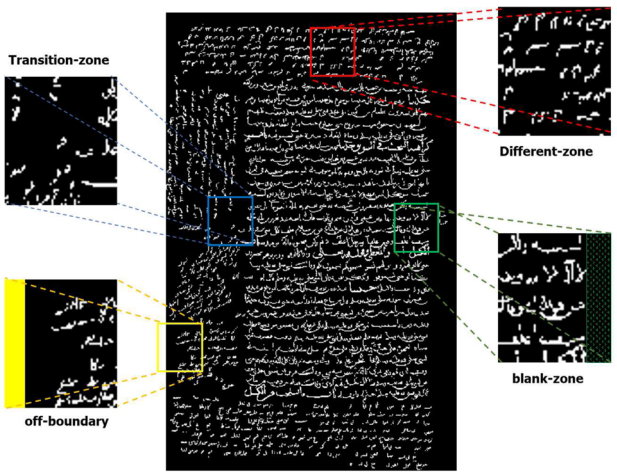


Fig. 8 Moving window stop conditions

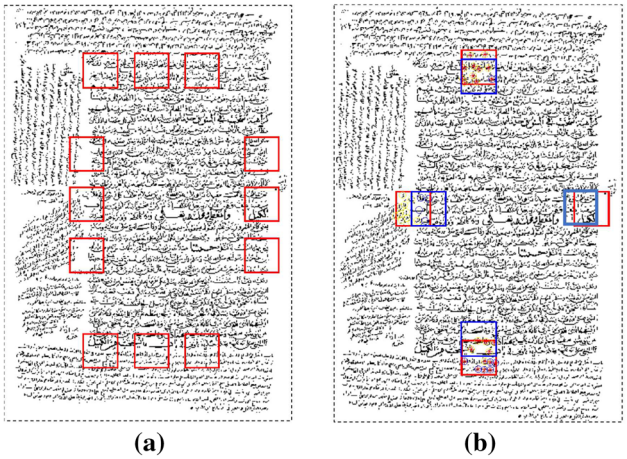


Fig. 9 Example of main-content boundary detection using moving-window algorithm; **a** boundary detection using moving windows, **b** stop correction

threshold th_2. Thresholds th_1 and th_2 are empirically computed in manuscript characterization step. Algorithm (1) outlines the main steps of the moving-window approach.

Figure 8 shows illustrative stop windows conditions. An example of moving windows analysis on a manuscript page is shown in Fig. 9a. In this example, only the right moving window was stopped because of a blank-zone condition, and all other windows were stopped because of the transition-zone condition.

Algorithm 1 Main-content boundary detection

Input: and a manuscript page I_c, Template Features F_{ini}
Output: Four stop windows $WinStops$

procedure MOVING WINDOW ANALYSIS
Loop in four directions (Right, Left, Up and Down):
Loop while not(off-boundary)
  Winfeat$_i$ ⟵ getNextWindow(I_c, step)
  $score$ ⟵ Match(Winfeat$_i$, F_{ini}, Euclidean)
  if ($score > th_1$) **then**
    FlagTransition ⟵ True
    return StopWin (Change in characteristics)
  else
    if ($score > th_2$) **then**
      FlagDifferent ⟵ True
      return StopWin (Different characteristics)
    else
      if ($score = \phi$) **then**
        FlagBlank ⟵ True
        return StopWin (Blank-zone)
Loop while (until off-boundary)
Loop next direction

3.3.2 Stop windows correction

Usually, stop windows require position correction to return main-content coordinates for final segmentation. In this

sense, the algorithm finds possible boundary cuts that separate main-content components from side-notes ones. The off-boundary and blank-zone conditions are corrected by moving back these stop windows toward main-content. Then, coordinates of boundary foreground pixels that are in the direction opposite to main-content are returned. In other words, if a window is stopped on the right boundary of main-content, then, it is corrected by moving it left and the rightmost foreground-pixels coordinates of the corrected window are returned. A similar procedure is carried out for other directions.

In Fig. 9a, left, upper, and lower windows are stopped because of the transition-zone condition. The correction of the transition-zone condition is more challenging. To address this issue, K-means clustering is used. We used six geometric features out of the mentioned eight features to perform K-means clustering, namely height, area, foreground/background ratio, pixel density, distance transform, and orientation. The K is set to two because we have two classes of writing main-content or side-notes. Finally, a cut is determined at the mid-distance between cluster centers as follows:

$$S_d = \frac{\text{dist}(Cn_i, Cn_j)}{2}$$

3.3.3 Main-content segmentation

The method selects boundary points from corrected stop windows to surround the main-content. Figure 10a shows an example of the selected boundary points. As the number of selected points is large, the method reduces them using a convex hull algorithm. Figure 10b illustrates the convex hull points in red color. After that, a convex mask is generated that surrounds the main-content. Finally, the manuscript's main-content is segmented by locating all connected-components that are completely within or they have 80% of their foreground pixels within the convex mask. Otherwise, connected-components are labeled as side-notes. Figure 10e shows an example of the main-content segmentation.

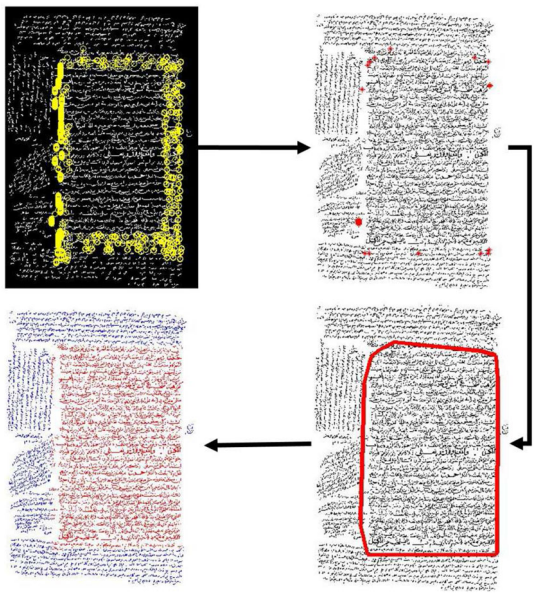


Fig. 10 Main-content boundary definition and segmentation

4 Experiments and results

Sine Asi et al.'s [4] approach considered main-content extraction from Arabic manuscripts; we compare our method's results to [4] method. There are three reasons for this comparison; (1) both methods are learning-free approaches, (2) the same Bukhari dataset is used to evaluate both methods, and (3) the analysis code of method [4] is publicly available in [1].

4.1 Datasets

Two datasets are used to evaluate the proposed method. The first dataset (referred later as DB1) contains 38 pages that were extracted from seven Arabic manuscripts. The manuscripts were scanned at a private library in the old city of Jerusalem [10]. DB1 pages contain two types of text main-content and side-notes written mostly by different writers. DB1 provides regional ground-truth images of the pages' main-content. The second dataset (referred later as DB2) contains 51 pages collected from the Islamic Heritage Project (IHP), available at Harvard online library [27]. The IHP contains 280 manuscripts in several subjects. We have selected pages that are mostly text and contain side-note text. Table 2 shows DB2 manuscripts' details. The DB2 ground-truth is manually created to have a similar representation as DB1.

4.2 Evaluation metrics

There are two main concerns in segmentation evaluation: type of correspondence and matching. Our aim in this study is to extract manuscripts' main-contents. Therefore, main-contents are considered for one-to-one evaluation correspondence. Second, a regional DLA matching can be computed by either the region's area or foreground pixels [16,26,43]. In this work, we used a pixel-based DLA matching.

The matching score is computed based on Pattern Recognition and Image Analysis (PRImA) Research Lab framework [2,3]. PRImA measure is used to evaluate DLA methods based on five segmentation errors:

Table 2 IHP dataset details

Transliterated title	Date	Subject	Samples
Kitab al-Taarruf li-madhhab ahl al-tasawwuf	10th century	Sufism doctrines	1–9
Kitab Bahr alkalam ilm al-tawhid	Undated	Islam doctrines	10–39
Bughyat al-talib marifat al-mafrud wa-al-wajib	1247 Hijri	Islam doctrines	40–51

– Merge: A segmented region overlaps more than one ground-truth region.
– Split: A ground-truth region is overlapped by more than one segmented region.
– Miss: A ground-truth region that is not overlapped by any segmented regions.
– Partial Miss (PMiss): A ground-truth region that is not completely overlapped by a segmented region.
– False detection: A segmented region overlaps no ground-truth region.

Based on these errors, a success rate SR for the segmentation can be computed. For each error type, the number of affected pixels is accumulated in ER [43]. Then, the success rate is calculated as follows:

$$\mathrm{SR} = \frac{\sum_{i=1}^{N} \omega_i}{\sum_{i=1}^{N} \frac{\omega_i}{1-\mathrm{ER}_i}} \tag{10}$$

where N is the number of error types, and ω_i is the final weight that is calculated for each error type as:

$$\omega_i = \frac{(N-1)\mathrm{ER}_i + 1}{N} \tag{11}$$

Also, standard Fscore is considered to evaluate our method and compare it to [4] method. The Precision (Pr) and Recall ($\Re$) are estimated as per Eqs. (12) and (13). True-positive (TP) is the rate of main-content PAWs labeled as the main text, False-positive (FP) is defined as the rate of side-note PAWs labeled as the main text, and False-negative (FN) is the rate of main-content components classified as side-notes.

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{12}$$

$$\Re = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

The Fscore is a single value that combines both the precision and recall. It shows how precise is the segmentation result to recall correct elements out of all segmented components. The Fscore is computed according to Eq. (14):

$$F\mathrm{score} = \frac{2 \times \mathrm{Pr} \times \Re}{\mathrm{Pr} + \Re} \tag{14}$$

Table 3 Performance evaluation using Fscore

Data	Avg Pr (%)	Avg Re (%)	Avg Fscore (%)
DB1			
[4]	97.94	84.30	90.11
Proposed	96.93	98.55	97.70
DB2			
[4]	98.67	95.27	96.78
Proposed	97.49	97.14	97.08

Table 4 Segmentation success rate

Data	Methods	SR (%)
DB1	[4]	70.41
	Proposed	98.5
DB2	[4]	92.97
	Proposed	97.1

4.3 Results and discussion

First, we reproduce the segmentation results of the [4] method using their provided MATLAB code [1]. Then, we evaluate the results of both methods using Fscore and success rate SR. Table 3 shows the performance evaluation using Fscore, and Table 4 presents the performance results in terms of segmentation success rate SR.

The proposed method shows promising results in terms of segmentation quality of main-content. Figures 11 and 12 show examples of segmentation results of the proposed method on DB1 and DB2 [in part (a)], and method [4] results are shown in part (b).

The results in Table 3 show the superiority of the proposed method in terms of Fscore using DB1. It achieved recall of $\Re = 98.55\%$ at $\mathrm{Pr} = 96.93\%$ precision. Moreover, the proposed method shows better segmentation results to extract main-content that reaches up to 98.5% success rate as indicated in Table 4.

By considering the success rate metric in Table 4, method [4] performance achieved a lower score of 70.41% using DB1. This low score is justified by analyzing the segmentation errors that are depicted in Fig. 14. Method [4] was suffering from Merge and Partial-Miss errors. Figures 11 and 12 show examples of the segmentation results on both datasets.

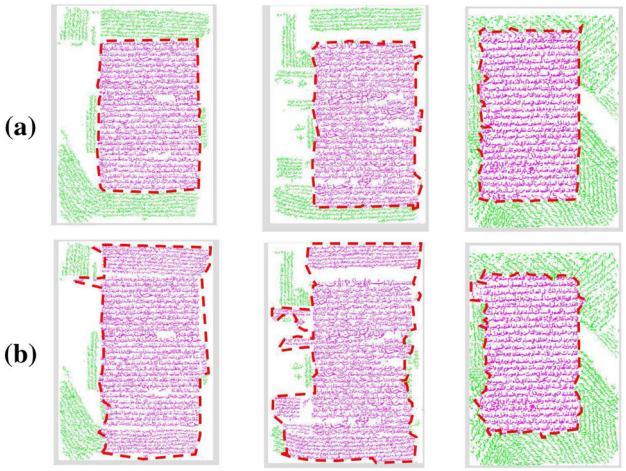


Fig. 11 Sample segmentation results using DB1. **a** The proposed method results, **b** segmentation results of [4] method

Fig. 12 Sample segmentation results using DB2. **a** The proposed method results, **b** segmentation results of [4] method

Comparing analysis time, our method is slower than [4] method because the proposed method is following a hybrid analysis strategy. It needs 159.5 s on the average for page analysis, while method [4] uses top-down analysis that requires 73 s on the average per page.

In the second experiment, the methods are evaluated using DB2. The performance of the proposed method is slightly degraded due to K-means clustering. In DB2, the amount of side-notes is small. Therefore, transition stops include a larger number of components from main-content than the ones from side-notes. In this case, K-means tends to generate two clusters that may contain main-content components in the second cluster. This leads to an increase in the segmentation of Partial-Miss errors. In contrast, the other method [4] shows better performance in comparison with its performance in the first experiment because the separation between main-content and side-notes in DB2 is much clearer.

There are some situations where DLA methods struggle to find the main-content boundary precisely. As discussed in Sect. 3, text-touching in a document with a dense writing style could be one major obstacle for any DLA method to produce accurate segmentation. The proposed method inherits the same limitation. Figure 13 shows the performance of the proposed method on DB1 and DB2 using a different number of frames.

To analyze segmentation degradation that has brought such performance, error analysis is conducted. Figure 14a and c indicates that most of the segmentation errors of the proposed method, on both datasets, are due to Partial-Miss errors. This type of error is happened due to how the proposed method defines the separating boundary. In other words, K-means lacks the precision to define the separating paths along main-content and side-notes. On the other hand, method [4] suffers high Merge error on both datasets (see Fig. 14b and d). Merge error reflects a limitation on how Gabor filter reacts to different texts written by the same writer or written in similar styles.

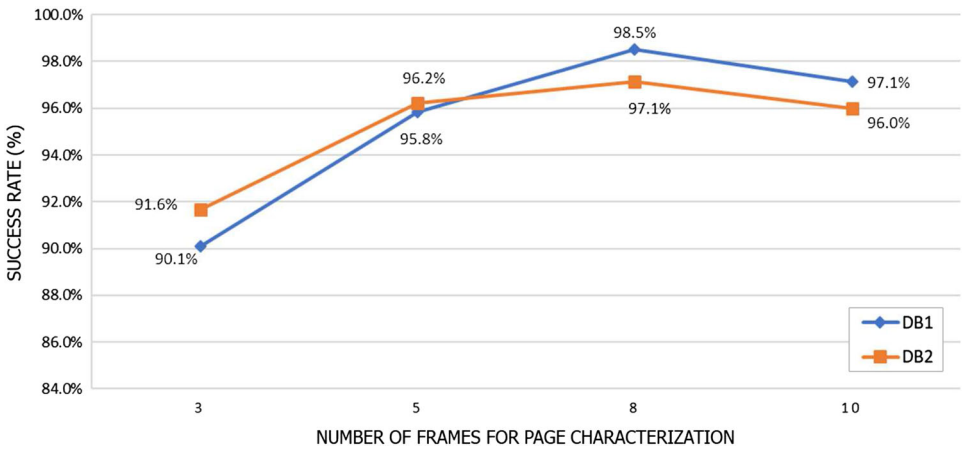
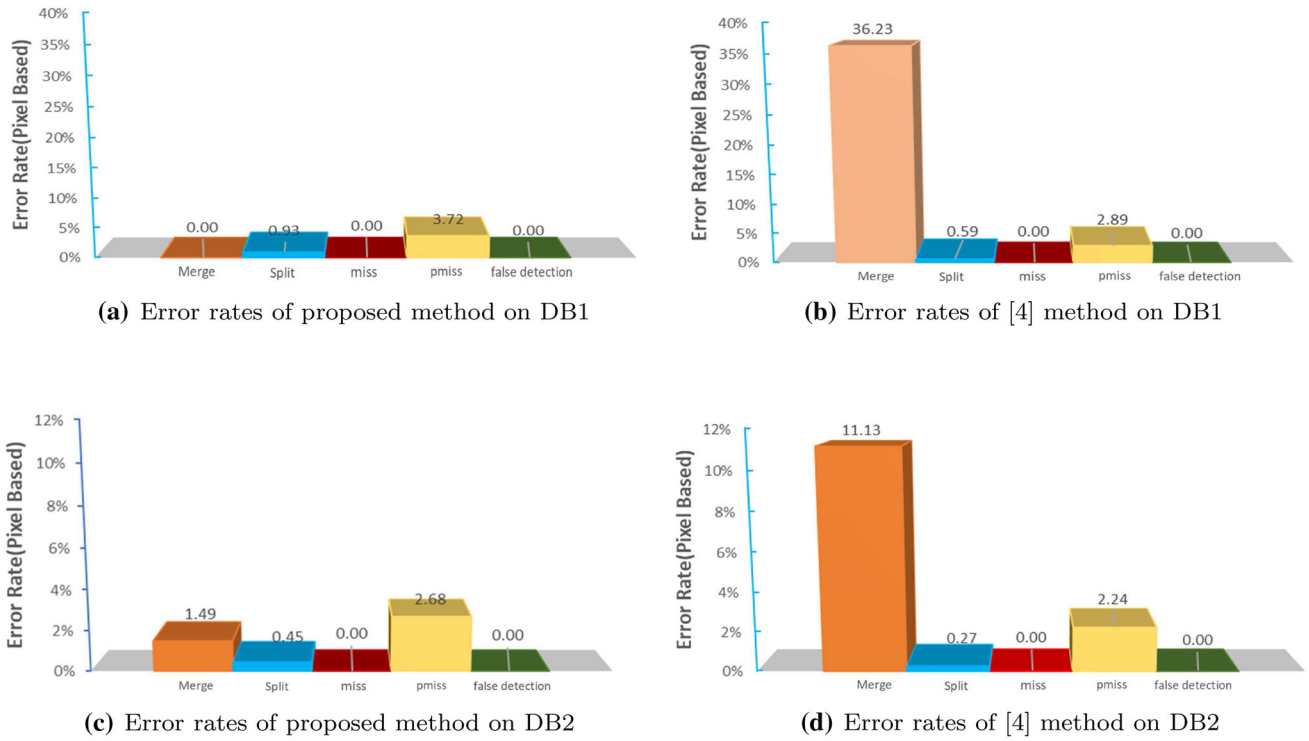


Fig. 13 Success rates of the proposed method



(a) Error rates of proposed method on DB1

(b) Error rates of [4] method on DB1

(c) Error rates of proposed method on DB2

(d) Error rates of [4] method on DB2

Fig. 14 Analysis error rates breakdown for the regional scenario; **a** error rates of the proposed method on DB1, **b** error rates of [4] method on DB1, **c** error rates of the proposed method on DB2, **d** error rates of [4] method on DB2

5 Conclusion

In this paper, a hybrid technique for handwritten Arabic historical manuscripts is presented. The aim is to extract the main-content from historical manuscript pages that could be used for document indexing and information retrieval. The proposed method considers its input uniquely, that is for every input page, the method extracts several features and parameters that help in document analysis. Anisotropic diffusion filtering showed faster localization of main-content. However, it may fail to determine the main-content boundary precisely due to text-touching. The moving-window analysis compromises this flaw at a cost of processing time.

In comparison with a learning-free method, the proposed method showed promising results that enhanced the segmentation of main-contents. Based on the error analysis of the proposed method, most of the segmentation flaws were due to Partial-Miss errors. For future work, text-touching between main-content components and side-notes should be addressed using context features. Moreover, we intend to expand the analysis method to detect side-notes text.

Acknowledgements The authors would like to thank King Fahd University of Petroleum and Minerals for the support during this work.

References

1. Abedelkadir, A.: Matlab code and dataset (db1). http://www.cs.bgu.ac.il/~abedas
2. Antonacopoulos, A., Bridson, D., Papadopoulos, C., Pletschacher, S.: A realistic dataset for performance evaluation of document layout analysis. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 296–300. IEEE (2009)
3. Antonacopoulos, A., Pletschacher, S., Bridson, D., Papadopoulos, C.: Icdar 2009 page segmentation competition. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 1370–1374. IEEE (2009)
4. Asi, A., Cohen, R., Kedem, K., El-Sana, J., Dinstein, I.: A coarse-to-fine approach for layout analysis of ancient manuscripts. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 140–145. IEEE (2014)
5. Baechler, M., Bloechle, J.L., Ingold, R.: Semi-automatic annotation tool for medieval manuscripts. In: 2010 12th International Conference on Frontiers in Handwriting Recognition, pp. 182–187. IEEE (2010)
6. Baechler, M., Liwicki, M., Ingold, R.: Text line extraction using DMLP classifiers for historical manuscripts. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1029–1033. IEEE (2013)
7. Baird, H.S.: The skew angle of printed documents. In: Proceedings of SPSE's 40th Annual Conference and Symposium on Hybrid Imaging Systems (1987)
8. Breuel, T.M.: Two geometric algorithms for layout analysis. In: International Workshop on Document Analysis Systems, pp. 188–199. Springer (2002)
9. Breuel, T.M.: An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis.

In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, pp. 66–70. IEEE (2003)
10. Bukhari, S.S., Breuel, T.M., Asi, A., El-Sana, J.: Layout analysis for arabic historical document images using machine learning. In: 2012 International Conference on Frontiers in Handwriting Recognition, pp. 639–644. IEEE (2012)
11. Bulacu, M., van Koert, R., Schomaker, L., van der Zant, T.: Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, pp. 357–361. IEEE (2007)
12. 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: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 299–304. IEEE (2016)
13. Chen, K., Seuret, M., Hennebert, J., Ingold, R.: Convolutional neural networks for page segmentation of historical document images. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 965–970. IEEE (2017)
14. Chen, K., Seuret, M., Liwicki, M., Hennebert, J., Ingold, R.: Page segmentation of historical document images with convolutional autoencoders. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1011–1015. IEEE (2015)
15. Chen, K., Wei, H., Hennebert, J., Ingold, R., Liwicki, M.: Page segmentation for historical handwritten document images using color and texture features. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 488–493. IEEE (2014)
16. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Scenario driven in-depth performance evaluation of document layout analysis methods. In: 2011 International Conference on Document Analysis and Recognition, pp. 1404–1408. IEEE (2011)
17. Corbelli, A., Baraldi, L., Balducci, F., Grana, C., Cucchiara, R.: Layout analysis and content classification in digitized books. In: Italian Research Conference on Digital Libraries, pp. 153–165. Springer (2016)
18. Cruz, F., Terrades, O.R.: Em-based layout analysis method for structured documents. In: 2014 22nd International Conference on Pattern Recognition, pp. 315–320. IEEE (2014)
19. Elanwar, R., Qin, W., Betke, M.: Making scanned arabic documents machine accessible using an ensemble of svm classifiers. Int. J. Doc. Anal. Recognit. (IJDAR) **21**(1–2), 59–75 (2018)
20. Garz, A., Sablatnig, R., Diem, M.: Layout analysis for historical manuscripts using sift features. In: 2011 International Conference on Document Analysis and Recognition, pp. 508–512. IEEE (2011)
21. Geusebroek, J.M., Smeulders, A.W., Van De Weijer, J.: Fast anisotropic gauss filtering. IEEE Trans. Image Process. **12**(8), 938–943 (2003)
22. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognit. **68**, 310–332 (2017)
23. Kang, L., Kumar, J., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for document image classification. In: 2014 22nd International Conference on Pattern Recognition, pp. 3168–3172. IEEE (2014)
24. Lam, S.W.: A local-to-global approach to complex document layout analysis. In: MVA, pp. 431–434 (1994)
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: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1096–1100. IEEE (2015)
26. Liang, J., Phillips, I.T., Haralick, R.M.: Performance evaluation of document layout analysis algorithms on the uw data set. In: Document Recognition IV, vol. 3027, pp. 149–160. International Society for Optics and Photonics (1997)
27. Library, H.: Islamic heritage project. http://ocp.hul.harvard.edu/ihp/scope.html
28. Maurer, C.R., Qi, R., Raghavan, V.: A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans. Pattern Anal. Mach. Intell. **25**(2), 265–270 (2003)
29. Mehri, M., Héroux, P., Gomez-Krämer, P., Mullot, R.: Texture feature benchmarking and evaluation for historical document image analysis. Int. J. Doc. Anal. Recognit. (IJDAR) **20**(1), 1–35 (2017)
30. Mehri, M., Nayef, N., Héroux, P., Gomez-Krämer, P., Mullot, R.: Learning texture features for enhancement and segmentation of historical document images. In: Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing, pp. 47–54. ACM (2015)
31. Nagy, G.: Twenty years of document image analysis in pami. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 38–62 (2000)
32. Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., Papamarkos, N.: Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image Vis. Comput. **28**(4), 590–604 (2010)
33. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
34. Ramel, J.Y., Leriche, S., Demonet, M.L., Busson, S.: User-driven page layout analysis of historical printed books. Int. J. Doc. Anal. Recognit. (IJDAR) **9**(2–4), 243–261 (2007)
35. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognit. **33**(2), 225–236 (2000)
36. Seuret, M., Chen, K., Eichenbergery, N., Liwicki, M., Ingold, R.: Gradient-domain degradations for improving historical documents images layout analysis. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1006–1010. IEEE (2015)
37. Seuret, M., Ingold, R., Liwicki, M.: N-light-n: A highly-adaptable java library for document analysis with convolutional autoencoders and related architectures. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 459–464. IEEE (2016)
38. Simistira, F., Seuret, M., Eichenberger, N., Garz, A., Liwicki, M., Ingold, R.: Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 471–476. IEEE (2016)
39. Simon, A., Pret, J.C., Johnson, A.P.: A fast algorithm for bottom-up document layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **19**(3), 273–277 (1997)
40. Singh, B.M., Sharma, R., Ghosh, D., Mittal, A.: Adaptive binarization of severely degraded and non-uniformly illuminated documents. Int. J. Doc. Anal. Recognit. (IJDAR) **17**(4), 393–412 (2014)
41. Singh, C., Bhatia, N., Kaur, A.: Hough transform based fast skew detection and accurate skew correction methods. Pattern Recognit. **41**(12), 3528–3546 (2008)
42. Tran, T.A., Na, I.S., Kim, S.H.: Hybrid page segmentation using multilevel homogeneity structure. In: Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, p. 78. ACM (2015)
43. Vasilopoulos, N., Kavallieratou, E.: Complex layout analysis based on contour classification and morphological operations. Eng. Appl. Artif. Intell. **65**, 220–229 (2017)
44. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Comput. Graph. Image Process. **20**(4), 375–390 (1982)
45. Wei, H., Baechler, M., Slimane, F., Ingold, R.: Evaluation of SVM, MLP and GMM classifiers for layout analysis of historical documents. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1220–1224. IEEE (2013)

46. Wei, H., Chen, K., Ingold, R., Liwicki, M.: Hybrid feature selection for historical document layout analysis. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 87–92. IEEE (2014)
47. Wei, H., Seuret, M., Chen, K., Fischer, A., Liwicki, M., Ingold, R.: Selecting autoencoder features for layout analysis of historical documents. In: Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing, pp. 55–62. ACM (2015)