Document Layout Analysis: A Comprehensive Survey

GALAL M. BINMAKHASHEN and SABRI A. MAHMOUD, King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia

Document layout analysis (DLA) is a preprocessing step of document understanding systems. It is responsible for detecting and annotating the physical structure of documents. DLA has several important applications such as document retrieval, content categorization, text recognition, and the like. The objective of DLA is to ease the subsequent analysis/recognition phases by identifying the document-homogeneous blocks and by determining their relationships. The DLA pipeline consists of several phases that could vary among DLA methods, depending on the documents' layouts and final analysis objectives. In this regard, a universal DLA algorithm that fits all types of document-layouts or that satisfies all analysis objectives has not been developed, yet. In this survey paper, we present a critical study of different document layout analysis techniques. The study highlights the motivational reasons for pursuing DLA and discusses comprehensively the different phases of the DLA algorithms based on a general framework that is formed as an outcome of reviewing the research in the field. The DLA framework consists of preprocessing, layout analysis strategies, post-processing, and performance evaluation phases. Overall, the article delivers an essential baseline for pursuing further research in document layout analysis.

CCS Concepts: • **Information systems** → **Document structure**; **Content analysis and feature selection**; *Information retrieval diversity*; *Document filtering*; *Information extraction*; • **Applied computing** → **Document analysis**;

Additional Key Words and Phrases: Document segmentation, document structure analysis, document image retrieval, document image understanding, layout analysis, physical document structure

ACM Reference format:
Galal M. BinMakhashen and Sabri A. Mahmoud. 2019. Document Layout Analysis: A Comprehensive Survey. *ACM Comput. Surv.* 52, 6, Article 109 (October 2019), 36 pages.
https://doi.org/10.1145/3355610

1 INTRODUCTION

Modern digitallibraries allow access to valuable/rare documents, especially ancient ones. Such libraries offer digital copies of the archived original documents online. These libraries enable fast ease of access to information. Usually, they use optical character recognition (OCR), which converts a document-image into text format. Consequently, with a simple text-based search, the requested information can be retrieved faster. On the other hand, some document-images are difficult to be converted to text-format perfectly, due to several issues such as bleed-through, faint ink, irregular writing styles, and the like. In this case, document image retrieval (DIR) is an alternative technique to search image archives using text queries without image-to-text conversion.

Authors' address: G. M. BinMakhashen and S. A. Mahmoud, King Fahd University of Petroleum and Minerals, Information and Computer Science, Dhahran, 31261, Kingdom of Saudi Arabia; emails: {binmakhashen, smasaade}@kfupm.edu.sa.

0360-0300/2019/10-ART109 $15.00
https://doi.org/10.1145/3355610

To get satisfactory information retrieval results, both applications, OCR and DIR, presume that some document layout analysis (DLA) has been conducted on those archived document images. In this sense, DLA is the first process in the pipeline of a document understanding system that detects and labels homogeneous document regions [79]. The early DLA methods analyzed simple document layouts and incorporated DLA as a secondary preprocessing task [105, 162]. Then, researchers encountered more complex document layouts that led to recognize DLA as a dedicated research field. [10, 13, 34].

There are two main aspects of document layout analysis that affect the advances in DLA research; the layouts diversity, and the evaluation metrics. Given document layouts diversity, a comprehensive and benchmark dataset, which spans most document layouts, is needed to facilitate document layout research. Although Antonacopoulos et al. [13] proposed a realistic dataset, they focused on magazines and technical/scientific documents only. Secondly, in the early stages of DLA development, many algorithms did not follow standard evaluation methods (i.e., subjective). Hence, the comparison of the researchers' results becomes difficult and inappropriate. Consequently, it negatively affected the progress in this research field [28, 148]. Recently, efforts are done by several researchers to standardize the DLA performance evaluation in several analysis context and objectives such as [8], [11], [15], [29], [97], and [151].

In this article, we provide a comprehensive survey of DLA algorithms by following a general DLA framework (see Figure 2). The current survey brings three major contributions to the research community. It proposes a comprehensive DLA framework and presents a critical study of DLA at the various analysis levels of the framework (i.e., regional analysis, text analysis, etc.); it summarizes the different DLA techniques in the literature and categorize them at the analysis strategy level; and then identifies and discusses the three levels of DLA performance evaluations. The proposed framework includes general DLA tasks that have been incorporated in various DLA methods. The survey discusses the main phases of DLA algorithms in detail; namely, preprocessing, layout analysis, and performance evaluation. Moreover, this study focuses on the physical structure analysis, and special concentration is given to historical document analysis algorithms (i.e., complex layouts). Historical document analysis algorithms are highlighted in this work, because most of the previous studies tend to review contemporary printed-document DLA [57, 98, 104], texture-based methods [112], skew-angle and document decomposition [39].

This survey is organized as follows: Section 2 presents the DLA framework. Section 3 discusses the DLA preprocessing phase that includes skew detection/correction and binarization methods. Section 4 discusses layout analysis, which includes analysis parameter estimation, page segmentation, and post-processing methods. The efforts on DLA evaluation and benchmarking datasets are presented in Section 5. Finally, our conclusions are given in Section 6.

2 DOCUMENT LAYOUT ANALYSIS FRAMEWORK

In this section, we present a discussion of a general document layout analysis framework. The DLA framework is constructed based on several reviewed studies that agree on common DLA phases. Moreover, as it is difficult to define perfect document layouts taxonomy, we identify several layout categories that suit both contemporary and historical documents.

2.1 Document Layout Types

Document layouts can be found in various structures. According to Kise [82], printed documents can be categorized into six types; rectangular, Manhattan, non-Manhattan, Multi-column Manhattan, horizontal overlapping, and diagonal overlapping. This categorization can be extended to historical manuscripts with arbitrary layouts. Since manuscripts are usually handwritten, Kise's categorization is also valid for contemporary handwritten document layouts.

Fig. 1. Document Layouts: (a) Regular, (b) Manhattan-based, (c) Non-Manhattan, (d) Multi-column Manhattan, (e) Arbitrary Complex, (f) Overlapping horizontally and diagonally.

The regular layout is characterized by large rectangular texts in single or multiple column/s with one paragraph on each column. Whereas, similar sources of documents with multiple paragraphs can be classified as Manhattan layout such as technical articles, magazines, official memos, and the like. The non-Manhattan layouts are those that have zones of non-rectangular shapes (see Figure 1(c)). The overlapping-layouts have some document elements such as texts that overlap other document-elements. Overlapping-layout may be due to show-through effect (see Figure 1(f)). These layouts can be classified as arbitrary (i.e., complex) when they are hand/typewritten using several styles, font types, and/or font sizes (see Figure 1(e)). In fact, there are several types of document-layouts, six of them can be designated as the most common. Moreover, the focus of this study is on techniques that extract text from documents leaving other types like drawings, maps, and the like intact. The other types can be processed as their regions are also specified. Figure 1 shows examples of the six most common document layouts. Document layouts are usually complex especially in historical manuscripts due to several factors; free-writing style, aging, faint text, ink bleeding, decorative text, and so on. In Figure 1, bold borders around regions describe the total shape that leads to our layout classification.

2.2 Analysis Framework

The variations in document layouts and the analysis objectives yield different DLA processing phases that vary from one algorithm to another. We observed a common workflow process among several reviewed DLA studies. This observation produces a general DLA framework as illustrated in Figure 2. The DLA framework consists of five phases; preprocessing, analysis parameter estimation, layout analysis, post-processing, and performance evaluation. A brief explanation of each phase is given below.

Preprocessing. Often, this phase is designed to transform an input raw document image into a method-oriented document image. In the other words, DLA methods may assume clean, binary, or de-skewed input images. Hence, the preprocessing phase should make sure that the input image meets the analysis pre-requirements. In general, this phase employs one or more of these essential preprocessing procedures such as binarization, de-skewing, and image enhancement.

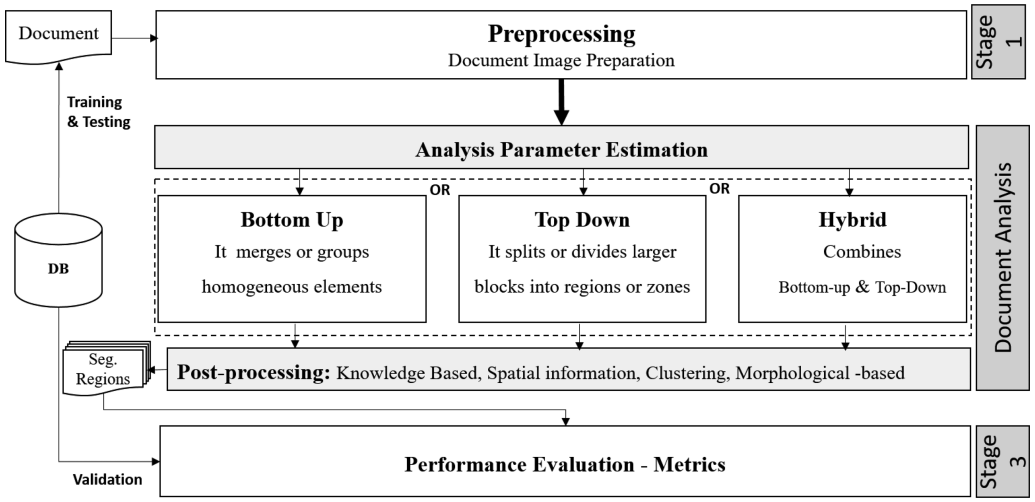


Fig. 2. General Document Layout Analysis (DLA) framework.

Analysis Parameters. They are pre-determined measurements that help DLA methods to control the document analysis. It can be divided into two types; model-driven or data-driven parameters. The model-driven parameters are estimated to fine-tune a DLA model to meet its analysis objectives. For instance, setting the number of nodes or layers of a Multi-layer Perceptron (MLP) or determining some initial weights for model training [34, 40]. On the other hand, data-driven parameters are computed using various measurements based on a given data set [19]. Examples of data-driven parameters are the average inter-line and word spacing, average height/width of characters, line-drawing size, and so on.

Layout Analysis. There are three types of layout analysis strategies; bottom-up, top-down, and hybrid. The bottom-up strategy often computes analysis parameters from the given data. It starts layout analysis at small document-elements such as pixels or connected-components. Then, it merges homogeneous elements to create larger zones. It continues forming larger homogeneous regions until it reaches pre-defined stopping conditions [98]. A top-down strategy starts from large document regions such as document-level. Then, it splits that large region into smaller zones such as text-columns based on some homogeneity rules. The top-down analysis stops when there is no more splitting of zones, or some stopping conditions are reached [105]. Finally, the integration of both strategies (bottom-up and top-down) yields what is called a hybrid strategy [74].

Post-Processing. The post-processing phase is an optional step in most DLA algorithms. Usually, it improves or generalizes the results of the DLA algorithm to other types of layouts [31]. It could be an essential step to compensate for any deficiency in the results to deliver accurate document segmentation as in [123].

Performance Evaluation. In general, document layout analysis has two main tasks; physical and logical analyses. The main purpose of the physical analysis is to detect a document structure and identify the boundaries of its homogeneous regions. On the other hand, the logical analysis is responsible for labeling these detected regions into document elements such as figures, headings, paragraphs, logos, signatures, and the like. The performance evaluation of a physical analysis method is performed using standard matching methods of segmented versus ground-truth entities either at the pixel or region levels. The DLA may deliver various forms of results based on

the type of analysis, physical or logical. There are some efforts to define a standard framework that allows coping with all possible results. For example, the Page Analysis and Ground-Truth Elements (PAGE) framework by Pletschacher and Antonacopoulos [119]. The PAGE framework was proposed to deal with the existence of a plethora of document representations and satisfies individual DLA stages. The ground-truth and segmentation results of DLA can be described using XML files. The description includes image-borders, the layout structure, page content, geometric distortions/corrections, binarization, and the like.

3 PREPROCESSING

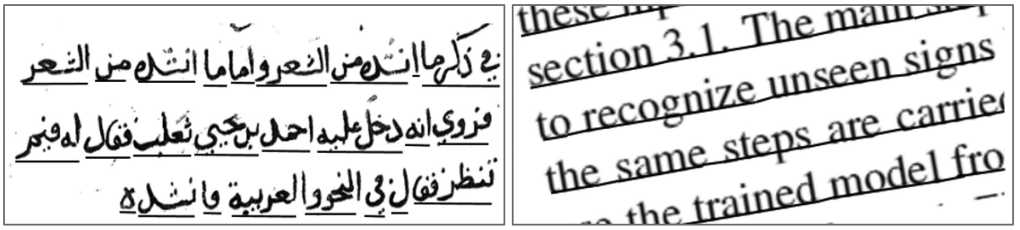
Document images may suffer from several degradations that negatively affect the performance of the DLA algorithms. Generally, there are two main sources of such degradations; native, and auxiliary [90]. The native degradation is generated due to aging, ink usage, writing style, etc. This type of degradation could lead to layout issues such as text ink-bleeding, show-through, text fading, text-touching, text-spacing or baseline fluctuation. Second, the auxiliary degradations are due to external factors such as a scanning-device malfunction, lighting conditions, and document alignment. Such factors may lead to document image skew, blurring, black-edges, and the like. The negative effect of these issues has to be minimized before starting any layout analysis. The preprocessing is an essential phase of many analysis algorithms. Although several studies did not emphasize their preprocessing procedures thoroughly, they assumed that the input images are ready for layout analysis. For the sake of giving a complete picture of DLA, we discuss in this section the classical preprocessing tasks, document skew correction and document binarization. This is because these preprocessing methods are considered frequently in the previous studies. Although document binarization is an important step in the pipeline of DLA, it has been neglected recently due to the advances in computer technology. The presence of powerful computing facilities allows many researchers to use full document-image information for document layout analysis using deep-learning methods. Usually, these DLA methods require complete pixel intensities to derive a better layout analysis [151, 167]. Hence, the binarization step is avoided in deep-learning DLA methods. Further details are given in Section 4.

3.1 Skew Detection and Correction

Document-image skew can be found at the level of document image (global) or at the level of document regions (local). Global document-image skew is formed because of auxiliary degradation. On the other hand, the writing style may cause local text skew . Both skew types require detection and correction before further analysis. Unlike document-image binarization, skew detection and correction is tightly related to document segmentation. In the other words, to extract document regions, the input document images should be set at a standard form (i.e., 0° skew angle). Several studies have discussed solutions to global and local document skew detection/correction such as [21], [145], and [161]. In general, previous methods can be grouped into seven categories: projection-profile, Hough transform, nearest neighbor, cross-correlation, line fitting, frequency domain, and gradient methods.

- *Projection Profile.* It is known for its ease of implementation and speed in the detection of text orientation [21]. Generally, it computes the sum of all pixel values along the horizontal direction (i.e., Assuming text is written horizontally) to form a vertical profile histogram. Then, the histogram is analyzed to find peaks and valleys (see Figure 3). In a typical setting, the peaks represent text-lines, and valleys represent line-gaps. However, in real situations, dense text, text-touching, and text-fluctuation can confound the projection profile histogram by several false peaks and/or valleys.

Fig. 3. Projection profile (vertical projection) on normal text lines.



(a) (b)

Fig. 4. Local Projection Profile Results; (a) Static stripes yields staircase effect [16, 175], (b) Dynamic strips [24].

For skew detection, several projection profiles can be computed over multiple orientations $[\theta_s - \theta_e]$ where $\theta_s = 0$ and $\theta_e = \pi$. Then, histograms of each projection are calculated along horizontal directions. Finally, the one that constitutes maximum variation indicates a document global skew-angle. This method can be applied at characters' level as in [23], and at the connected components' level as [106]. The application at fine levels requires large computations because of the repetitive calculations of the projections at each direction. Although image resizing could lower the number of computations needed, the method accuracy will also be affected.

In general, the projection profile methods suit printed documents. It may fail to find the true skew angle when there are several non-text regions in a document image [136]. Preprocessing a document image and removing non-text regions may improve the projection profile detection [88].

Finally, to boost the performance of the projection profile approach against arbitrary layouts, it can be applied locally either statically or dynamically [16, 24, 175]. In the local static approaches, the algorithm divides a document image into fixed vertical stripes and applies the projection profile approach on every stripe [16, 175]. Such static application treats each stripe exclusively, which yields a staircase effect (see Figure 4(a)). On the other hand, dynamic local projection-profile defines stripes with overlaps [24]. Therefore, no staircase effect will be generated in the output. Figure 4(b) shows the dynamic local projection-profile results that allow tilted text-line analysis.

- *Hough Transform.* The basic idea of Hough transform is to perform angular scanning of image pixels and accumulate votes of each scan in Hough space [107]. Then, a candidate

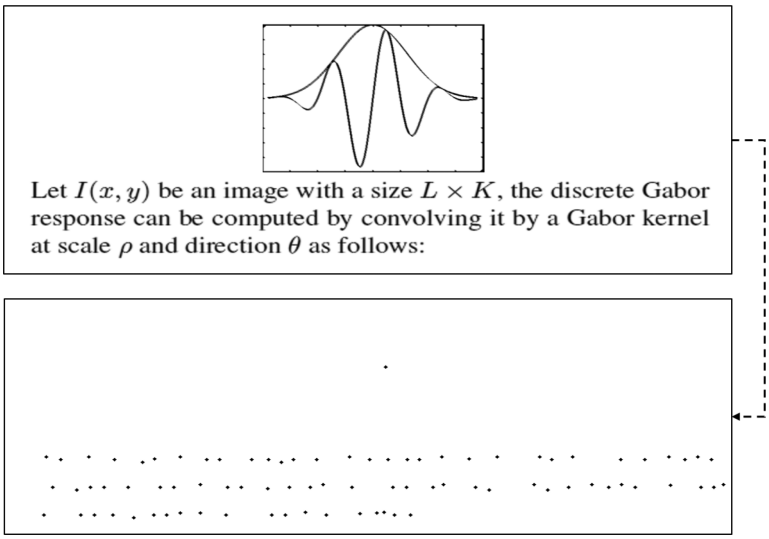


Fig. 5. An example of data reduction before applying Hough Transform.

line is detected by finding the highest response in the Hough space. The Hough transform was used in detecting documents skew for several years [115, 154, 173].

Some researchers used preprocessing to improve the document-images before applying Hough transform. For example, smearing text-lines is used to overcome characters in-between spaces and yield line drawings for every text-line [73], or shrink characters to ink-points as in [87], [89], [92], and [172] (see Figure 5). Furthermore, edge detection methods were used to expose line-structure of text-lines [107]. Figure 5 shows an example of character shrinking method. The text shows regular dots along the horizontal/vertical directions, while the figure components have larger vertical gaps. After such preprocessing, Hough subspace is generated by scanning the processed document image. Then, the generated subspace is searched to find angles (θ) with the highest responses. The average of these angles describes the global document skew-angle [154].

Although Hough-based techniques can detect skew angle accurately with a small error margin (see Table 1), they require high computation to preprocess an image and build the Hough subspace. Moreover, the line structure of a handwritten text may be hard to be computed especially in historical documents.

- *Nearest Neighbor Approach.* The distance relationships among connected components can be utilized to detect and correct document skewn. First, this technique divides a document image into small components. Then, it finds all relative neighbors along specific directions. After that, it accumulates the angles of these components in an angular histogram where the peak value indicates a document's skew-angle [111]. Although the nearest neighbor approach is applicable to various document layouts, the resultant skew angle estimation may lack precision [111]. Like Hough-based approaches, handwritten-documents skew detection/correction is challenging. Therefore, most of the studies on the nearest neighbor skew detection/correction are performed on printed documents such as [6], [58], [91], [95], [135].
- *Cross-Correlation.* Cross-correlation method analyzes text lines to detect/correct the skew of document images. Yan [171] accumulated foreground pixels across pairs of inter-text line spaces. Yan's method moves horizontally and then vertically with two different fixed

shifts d_h, and d_v, respectively, to compute the cross-correlation as in Equation (1):

$$R(d_v) = \sum_{x=0}^{X-d_h-1} \sum_{y=0}^{Y-d_v-1} I\,(x+d_h, y+d_v) \times I(x,y) \tag{1}$$

where $X-1$ and $Y-1$ are the image width and length, respectively, and $I(x,y)$ is the image pixel at location x, y. The maximum of $R(d_v)$ indicates an optimal d_v value of an average vertical gap. Thus, the skew angle is computed by dividing the optimal vertical gap by the difference of the two horizontal shifts (Δd_h):

$$Skew = \tan^{-1}\left(\frac{d_v}{\Delta d_h}\right) \tag{2}$$

It can be observed that Yan's method requires a static inter-text line spacing. Alternatively, the cross-correlation can be applied on equidistant vertical lines of a document image [66]. In this way, the method does not rely on inter-text spacing and considers less image pixels to process. Generally, the cross-correlation methods are limited to document skew within $\pm$ 15° skew angles (see Table 1).

- *Line Fitting.* Like Hough-based approaches, line-fitting methods by nature do not require large input to find text lines. Only two points can be used to describe a line segment. For instance, the method divides text into connected components and represents each connected component using a single point called Eigen-point. Then, the coordinates of these Eigen-points are used as input to a linear regression estimator and detect document skew-angle [37].

  Although reducing connected components to Eigen points has shown to be better than centroids [37], a single point representation for a connected component is not enough due to fluctuated text-lines. Shivakumara et al. [150] proposed a line-fitting approach using two-point representation; uppermost and lowermost Eigenpoints. This method was compared to studies that used single-point representation and it showed to be superior in text-line fluctuations.
- *Frequency Domain.* Postl [120] method is among the earliest approaches that applied Fourier transform (FT) to detect/correct document image skew. A modified version of Postl's algorithm was proposed by Peake and Tan [116]. Peake and Tan's method determines a document skew angle by accumulating local document-blocks' angles into an angular histogram. Although Peake and Tan's method enhances the computation cost, it loses some accuracy due to inconsistent FT spectrum from various blocks. This issue was addressed by normalizing the FT local responses [94]. Lowther et al. [94] suggested Radon transform to analyzed the FT spectrum. Figure 6 illustrates FT and Radon transform example of document skew angle detection. In Figure 6(c), Radon transform shows several peaks due to minor responses on the Fourier spectrum. The minor responses were caused by irregularities in text and drawings. This issue was addressed by clustering these irregularities with the nearest content to generate convex shapes. Then, Radon transform was applied on the resultant convex shapes that represent document's blocks to compute document skew [58].

  Radon transform and energy function analysis were integrated to calculate document skew in [61]. Radon transform is similar to the projection profile approach which scans the document in multiple orientations. Consequently, it inherits the projection profile's limitations. For this reason, the method in [61] divided the document image into blocks. Then, it used a bootstrap aggregating (Bagging) to accumulate blocks' local skew-angles to compute the final document skew angle. Another study suggested an integration of Radon and wavelet transforms to compute document skew [3]. It considered only text components to

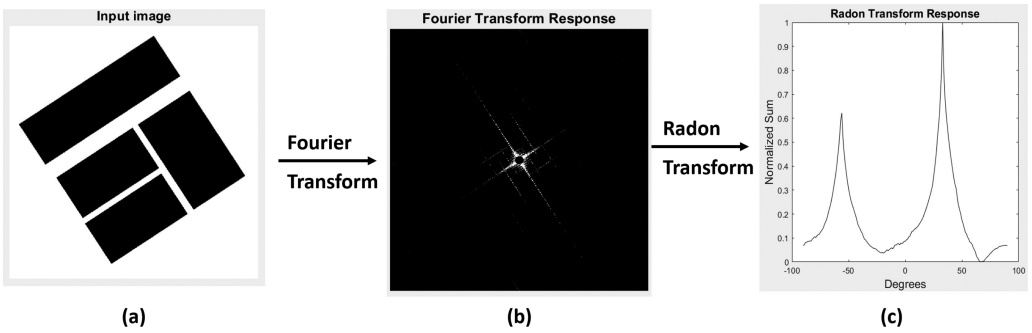


Fig. 6. Frequency domain analysis: (a) Original document image, (b) Fourier Transform Magnitude, (c) Radon transform.

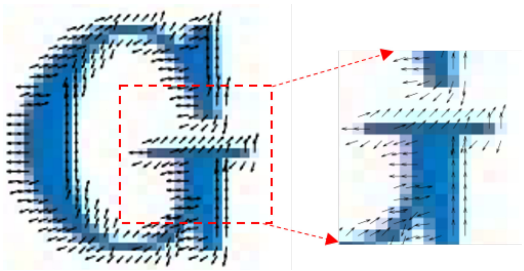


Fig. 7. Confusion of gradient performed on letter "G" [53].

compute the document skew and removed all non-text components before detecting the skew-angle.

- *Gradient Approach.* Similar to other methods, gradient methods require conversion of characters into edges or lines before carrying out skew detection/correction [138]. Usually, text components have many edges and corners that represent writing strokes. Therefore, gradient approaches could produce responses in many directions and confuse the skew detection. Consequently, it requires letter-based preprocessing to avoid false detection of document skew. Figure 7 shows an example of the gradient magnitude and angles (represented as arrows) on the letter "G".

  Sauvola and Pietikïnen [138] suggested to collapse the letters' structures by reducing image resolution by a factor of 1/5. Then, their algorithm determines the document skew angle by computing an angular histogram maximum point. The angular histogram A_θ^w is computed using Equation (3):

$$A_\theta^w = \sum_{(i,j)\in W} Mag(i,j)\cos^2(\theta - \emptyset_{ij}) \tag{3}$$

  where θ is a rotational angle from 0 to 179, Mag and $\emptyset_{ij}$ are the magnitude and phase of the gradient at position (i, j), respectively.

  Another gradient method that addressed angle confusion due to direct application of gradient approach is described in [53]. Diem et al. [53] method combines gradient algorithm with focused nearest neighbor clustering of interest points to calculate the document skew.

Summary

Table 1 summarizes the discussed document skew detection/correction techniques in this section. Comparison of these approaches is difficult and could be biased because each work used different

Table 1. Skew Detection and Correction - Summary Table

Md	Ref	Angle		Document				Md	Ref	Angle		Document			
		Rng	Err	Typ	TS	Lng	Lyt			Rng	Err	Typ	TS	Lng	Lyt
PP	[24]	±45°	0.2	H	30	Multi	MB	Gnt	[53]	≤180°	1.75	H	658	Eng.	MB
	[136]	≤25°	0.1	P	3	Multi	MB		[138]	≤20°	0.1	P	11	Eng.	MC
	[88]	±15°	0.2	P	500	Eng.	MC	NN	[58]	NA	0.05	P	175	Eng.	MC
	[16]	NA	0.12	H	720	Multi	MB		[135]	≤40°	0.33	P	979	Eng.	MC
	[21]	≤360°	0.1	P	270	Eng.	MC		[6]	≤180°	NA	P	30	Multi	MB
	[175]	NA	0.3	H	100	Arb.	MB		[95]	±45°	0.2	P	280	Multi	MC
	[106]	≤40°	0.1	MX	8	Multi	MB		[91]	±45°	−6	P	78	Eng.	MB
	[120]	±45°	0.6	P	NA	Eng.	MB		[111]	≤180°	NA	P	NA	Eng.	MC
HT.	[55]	≤180°	0.3	P	500	Eng.	MC	CrC	[66]	±4°	0.068	P	NA	Eng.	MC
	[107]	2–20	0.07	P	20	Eng.	MB		[171]	≤12.5°	NA	P	2	Eng.	MC
	[154]	≤150	0.05	P	300	Eng.	MB	LF	[150]	≤30°	0.5	P	100	Eng.	MB
	[173]	≤180°	0.1	P	NA	Eng.	MB		[37]	4.2°, 3.8°	0.029	P	200	Eng.	MB
	[115]	≤45°	0.84	P	100	Eng.	MB	FD	[3]	1-25	NA	P	150	NA	MB
	[73]	±45°	NA	P	13	Eng.	MC		[80]	≤120°	2	P	100	Arb.	MC
	[87]	±15°	0.167	P	250	Eng.	MC		[94]	±45°	0.25	P	94	Eng.	MC

[a] Rng: Range, Err:Possible Error, TS: Testing Samples, Lng: Language, Lyt:layout, Md: Method.
[b] PP: Projection Profile, HT: Hough Transform, NN: Nearest Neighbor, CrC: Cross-Correlation, LF: Line-Fitting, FD:Frequency Domain, Gnt: Gradient-based.
[c] P: Printed, H:Handwritten, MX: mixed type documents, Eng: English, Arb: Arabic, MB: Manhattan based, MC: Multi-column document, NA: Not Applicable.

test documents. However, the summary table may help in identifying the methods capabilities and working angle ranges. To sum up, the projection profile, and frequency domain techniques are able to detect document skew in large skew-angle range. However, they may be severely affected by non-text contents. Hough transform, line-fitting, nearest neighbor, and gradient-based techniques have mid-ranged skew correction ability up to 180°. On the other hand, they inherited noise sensitivity and may require document-image cleaning. The cross-correlation techniques are limited to correct document-image with ±15° skew.

3.2 Document Image Binarization

In general, binarization process converts a given grayscale image into a binary image using pre-computed thresholds. Commonly, the produced binary image contains a value of one for background pixels and zero for foreground pixels (and vice-versa). Binarization is still an important phase in the document understanding pipeline. Usually, binarization assists the subsequent analysis phases in addressing several analysis tasks such as text-line detection, skew correction, and connected-component estimation. Other application areas include the restoration of historical document-images and verification of signatures. It is still considered a major research area because it helps in analyzing complex document challenges such as noise, complex background, uneven illumination, and faint foreground that is often caused by smearing, bleeding through, blurring, aging factors, and so on. Recently, more than 25 binarization methods were submitted by 18 research groups in the 2017 Document Image Binarization Contest (DIBCO) series [122]. In comparison to previous versions of DIBCO contests of 2016 [121] and 2014 [110], the participation nearly doubled each year.

The binarization process reduces the required analysis computations to approximately one-third because only one channel will be considered as an input to the next phase. Moreover, it

automatically overcomes various layout issues such as text show-through, blurring, and the like. Typically, a binarization process computes a threshold that is used to classify pixels into either the foreground or background. In this sense, we can categorize the binarization methods based on how a binary threshold is being computed? Four types can be identified; pixel variance, entropy, contrast, or error-minimization thresholding methods.

The variance-based binarization methods estimate the optimum grayscale threshold that separates foreground from background pixels at a point where the intra-class variance is minimal. Otsu [114], Niblack [108], and Sauvola and Pietikäinen [139] are examples of this type. However, due to the large noise (i.e., pixel outliers) in historical document-images, the variance-based or entropy-based methods' performance can be degraded. Therefore, contrast-based binarization approaches were suggested [26]. The contrast approaches can overcome outliers noise by considering the contrast information of the document-image blocks dynamically. Finally, binarization can be formulated as a classification problem where it computes a global threshold that minimizes the classification error rate [72]. Recently, machine learning methods for document image binarization are proposed [122]. For instance, He and Schomaker [72] showed that with an application using deep-learning for document image enhancement, a simple global binary thresholding (i.e., Otsu) can yield the state-of-the-art binarization results (it is called DeepOtsu method). Moreover, Westphal et al. [166] proposed a Recurrent Neural Network (RNN) to compute the document image binarization locally.

In general, binarization can be applied at either global or local levels. The global-based methods compute the binary thresholds over the document image holistically [114]. On the other hand, local-based methods compute several thresholds based on the local characteristics found in the current processing part of an image. The scenario of dividing a document-image into a set of small image patches before applying binarization is an example of local binarization application. However, such application may not be robust to noise especially if the patches suffer form large contrasts that vary notably from one block to another. This issue was addressed using an adaptive binarization in [46]. Usually, The adaptive methods use small image patches with overlaps among them. Therefore, these methods estimate binary thresholds for the current patch by considering its neighbor patches' information. Examples of adaptive binarization are [7], [46], [49], [68], and [127].

Although adaptive binarization methods showed robustness to contrasts and outliers, they are still not perfect. For instance, Gatos et al. [68] concluded that using adaptive binarization on images that suffer from severe illuminations may negatively impact the produced binary image. The illumination imbalance usually introduces brighter and darker image pixels that perplex the binarization method. Therefore, it is recommended to conduct some image preprocessing operations such as image equalization, and smoothing to overcome these issues [68, 153].

The dark foreground pixels in writing transition zones are another issue that binarization methods are facing. Su et al. [155] proposed normalizing each image patch to minimize the effect of dark pixels. Then, it computes a statistical threshold for each image-patch by considering only the transition-pixels. Although such method showed robustness in preserving text edges, it critically depends on two empirical parameters such as the neighborhood window-size.

In summary, the binarization of document-images that suffer from varying illumination and noise is a challenging task [41, 122]. For instance, deformations in text shapes such as fractures and merges due to dark-pixels or illumination can severely affect the threshold estimation [110]. Document image binarization is still imperfect and open for research, despite several studies that have integrated multiple techniques such as [68] or that performed document image enhancement before binarization such as [72], and [155].

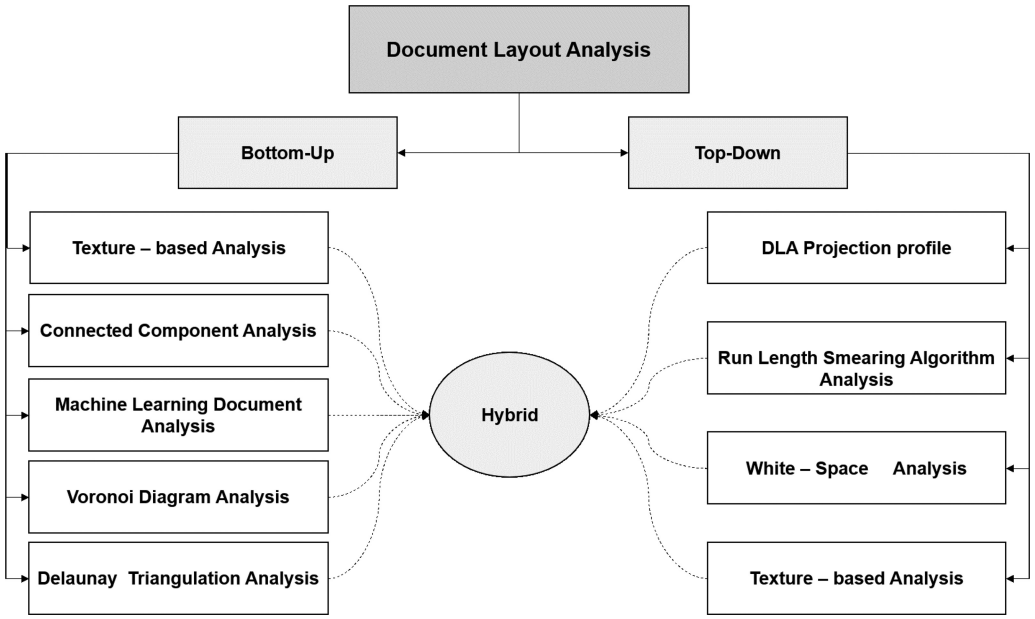


Fig. 8. Document layout analysis taxonomy.

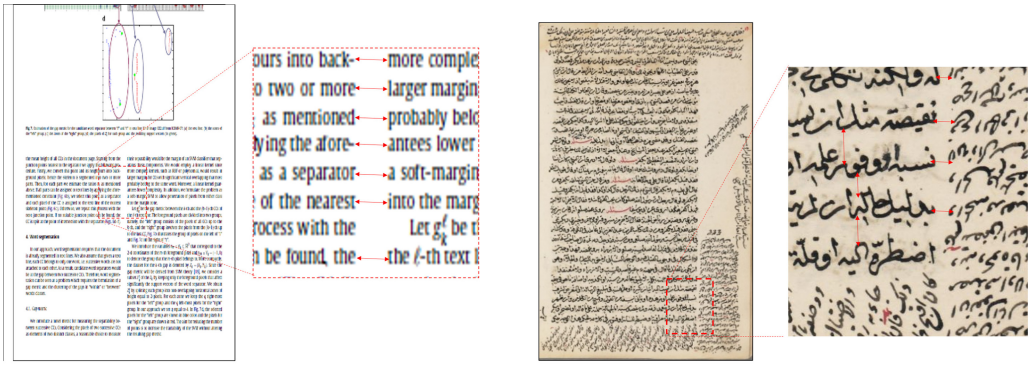
4 DOCUMENT LAYOUT ANALYSIS

In this section, we present the core tasks of document layout analysis; parameter estimation, page segmentation (i.e., analysis), and post-processing. On one hand, The parameter estimation and post-processing tasks are discussed briefly due to two reasons; (1) they are optional tasks of DLA, and (2) usually, they are discussed with minimal details. On the other hand, we shed more light on various document layout analysis methods. The discussion of methods are mapped to the classical DLA strategies; bottom-up, top-down, and hybrid. Figure 8 shows a general taxonomy of the document layout analysis methods.

4.1 DLA Parameter Configuration

Most of the DLA algorithms, either bottom-up or top-down, require analysis parameters to identify the different regions. It is difficult to categorize analysis parameters used in the previous studies. However, the analysis parameters are either set at the beginning of the analysis (static) or can be changed during the analysis (dynamic) [169]. In general, these parameters are critical thresholds that should be set carefully to perform robust DLA.

Static parameters can be determined at the beginning of a document analysis. They remain fixed throughout all processed documents. The static parameters suit DLA algorithms that analyze constrained (i.e., structured) document layouts. Examples of a static parameter estimation are text-blocks locations [162], regular region-gap size [105], regular number of lines per region [134], and the size of text elements [131]. On the other hand, the dynamic parameter estimation (i.e., data-driven) is computed from a document image directly. This type of estimation is used whenever the documents under analysis are heterogeneous. The parameters are dynamic because they change from one document to another. Examples of such methodology can be found in [75], [140], and [168]. Figure 9(a) illustrates an example of possible static parameter estimation such as regular font size. Figure 9(b) shows a situation where a dynamic parameter estimation should be used because of variable writing styles.



(a) Regular spacing and writing styles (b) Variable spacing and writing styles

Fig. 9. Parameter estimation: (a) Static parameters, (b) Dynamic parameters.

4.2 Bottom-Up Strategy

Usually, bottom-up strategy derives document analysis dynamically from smaller granularity data levels. It estimates the parameters using statistics of pixel distributions, properties of connected components, words, text lines, or regions. In general, bottom-up analysis starts at fine levels of an image such as pixels, components, or words. Then, the analysis grows up to form larger document regions and stops once it reaches a predefined analysis objectives. In this subsection, we discuss the bottom-up strategy based on five core categories; namely, connected component analysis, texture analysis, learning-based analysis, Voronoi diagrams, and Delaunay triangulation.

4.2.1 Connected Component Analysis. The connected component analysis allows more flexible layout analysis because it offers a wide range of shape properties. Docstrum algorithm is among the earliest successful bottom-up algorithms that is based on connected component analysis [111]. It groups connected components (CC) on a polar structure (distance and angle) to derive the final segmentation. Even though Docstrum can cover a wide range of layouts, it was tested on printed documents. Furthermore, the local features of the connected components have helped researchers address some historical manuscripts layout issues [30, 31]. Another work by Rabaev et al. [125] proposed evolution maps of connected components on grayscale and binary versions of a document image to extract degraded text lines.

In general, connected-component-based layout analysis requires feature extraction and machine learning methods [158]. For example, Tran et al. [159] proposed an iterative classification method that uses connected components to differentiate four classes of document's components as figures, separators, text, and noise. In each iteration, the method removes any connected component from a block found to be heterogeneous to its neighbors. The connected component is heterogeneous if it is non-text. This process is continued until all regions become homogeneous (i.e., only text). For other non-text regions, the algorithm uses the connected component geometric properties to detect figures, separators, and noises.

4.2.2 Texture Analysis. Texture analysis enjoys speedy detection of document-image elements. Texture analysis techniques can be categorized as bottom-up or top-down based on how a method approaches a document layout analysis? In other words, texture analysis has a wide range of methods that can be classified as bottom-up or top-down.

Usually, in bottom-up texture analysis, it starts by extracting texture features directly from the image pixels. Then, these features are used to group pixels to form homogeneous regions. For instance, spatial autocorrelation approach is one example of a bottom-up texture-based DLA

[76, 77]. This algorithm auto-correlates the document image with itself to highlight periodicities and texture orientation. Finally, the texture orientations are analyzed in a directional rose. In the rose-of-directions diagram, text strokes are highlighted by thin response, while thicker responses represent graphic elements. This behavior of the rose-of-directions has been utilized in document segmentation [76].

Journet et al. [77] algorithm is computationally expensive due to computing features to cover multiscale using repetitive resizing of a complete document-image. Instead, a moving window should be resized to achieve multiscale features and lower the computational requirements.

Due to a successful application, the autocorrelation approach is compared to other texture analysis techniques such as Gray Level Co-occurrences Matrix (GLCM) and Gabor filter bank [101]. The authors concluded that Gabor filter is suitable to distinguish textual regions provided that distinct font or similar writing style was used, while the autocorrelation approach is better if a document possesses a complex layout or its text are written in different fonts.

Texture analysis that works directly on the pixel level is computationally expensive. Mehri et al. [103] suggested a DLA based on superpixels. A superpixel is a group of pixels that shares similar spatial and intensity information. Although the superpixeling step could leverage the separation between the foreground and background and boost the layout analysis, it increases overall analysis time. Finally, a comparative study of nine types of texture feature extraction methods is reported in [102]. Their study concluded that Gabor texture is the best choice among the tested nine techniques to distinguish textual content from the graphical ones. Furthermore, they can distinguish different fonts effectively.

4.2.3 Machine Learning Document Analysis. Machine learning methods can be viewed as either top-down or bottom-up methodologies. Since this approach uses either direct pixel-intensities or pixel-features to identify zones and regions, the machine learning DLA methods are categorized as bottom-up in this study.

There are several DLA studies that considered machine learning to address various document understanding issues including preprocessing, segmentation, and labeling tasks [99]. Various machine learning algorithms were adopted in DLA such as radial basis function network, probabilistic neural network, and Self-Organizing Maps [149], time delay neural network [100], space displacement neural network [99], and support vector machines [163]. However, the multilayer perceptron (MLP) architecture and learning scheme are the most dominant Artificial Neural Network (ANN) style used in the literature [99]. In the following discussion, the machine-learning methods are further divided into non-deep, and deep learning methods.

- Non-deep learning methods
  Non-deep learning methods use simple neural network architectures to learn machine models for DLA. The analysis using ANN is conducted at three levels; pixels, block, and page. Direct pixel intensities may not be the best choice to build a conventional machine learning model in comparison to feature-based. Data imbalance and missing context information are the main issues that the learning-based methods suffer from. For instance, given a document as input data for model training, usually textual or background data are much larger than line-drawings or logos data. Consequently, a trained model could be biased towards text or background pixels [22] and [71]. A dynamic MLP (DMLP) was proposed to learn a less-biased machine model using pixel-values and context information [22]. The dynamic MLP network is not fully connected to reduce the influence of data imbalance in machine model training. Another issue that arises when using pure-pixels in machine learning DLA is losing context information [45] and [59].

Usually, block and page-based ANN analysis require features extraction methods to empower the ANN training and build robust models. These features can be either handcrafted or generated automatically. The handcrafted features are developed through feature extraction techniques such as Gradient Shape Feature (GSF) [52] or Scale Invariant Feature Transform (SIFT) [62, 63, 65, 164], to name a few. Garz et al. [63] found that SIFT interest points are usually scattered around text regions. Hence, it can be used for a text-line extraction task. There are several other techniques that use feature extraction methods such as texture features [44, 101–103, 163, 165], geometric features [30, 31].

- Deep learning-based methods
  Recently, machine learning algorithms became dominant in solving pattern recognition and computer vision problems. This is due to the vast advances in computer technology that allowed fast processing and support larger memory capacities. Consequently, the learning-based DLA methods get more attention to address complex layout analysis and derive both the logical and physical layout analyses [123].

  Moreover, the effect of data imbalance can be reduced using weights such as [38]. Capobianco et al. [38] suggested a Fully Convolutional Neural Network (FCNN) with a weight-training loss scheme. The method was designed mainly for text-line extraction, where the suggested weighting loss in FCNN has helped in balancing the loss function between the foreground and background pixels. Another study by Chen et al. [42] suggested automatic stacked convolutional autoencoders to learn features from superpixels for document layout analysis. The grouping of pixels to superpixels has reduced the effect of data imbalance.

  Unlike conventional machine-learning methods, usually deep learning methods generate features from image-pixels for document layout analysis. However, most of them may require post-processing of the results. For instance, Wick and Puppe [167] proposed an FCNN method, which consists of five encoders and three decoders, for document layout analysis. Their method requires a binarization step to keep track of the document's foreground pixels in the preprocessing phase. Then, it uses the same binary mask in post-processing the final FCNN segmentation results. Another work by Grüning et al. [70] proposed ARU-Net for DLA. ARU-Net is an extension of the U-net [130] that considers special attention (A) and depth residual structure (R) to overcome the pooling issue of the previous deep learning methods. The segmentation results of this complex deep network were post-processed using a clustering algorithm to identify text-lines. Fortunately, a simpler deep-learning architecture was tested for document layout analysis in [43]. Chen et al. [43] suggested a simple FCNN architecture that contains one convolution layer for page segmentation. Their results were comparable to complex deep networks.

  Besides the required post-processing of outcomes of deep learning methods, a network parameter initialization is another main concern. Usually, deep networks require huge data to learn important data parameters for segmentation or classification tasks. Therefore, the training process takes long time and large computation requirements. In general, there are three approaches for machine-learning training initialization; random weights, transfer learning, or unsupervised layer-wise pre-training initialization [141]. Almost all conventional neural networks are usually initialized using random weights. Consequently, these trained models may be trapped in some local minima. Often, it is necessary to repeat the training for several epochs with different initialization weights to avoid such local minima.

  Other machine learning methods may use weights of pre-trained networks. The pre-trained networks provide faster learning convergence. A study by Oliverira et al. [113] proposed a multi-task document layout analysis approach using Convolution Neural Network

(CNN). The method adopted transfer learning using ImageNet [51]. The ImageNet was used as a deep residual network followed by five contracting steps to reconstruct segmentation results.

A layer-wise network initialization is an alternative that derives network parameters from the targeted data to speed-up model training and stabilizes performance accuracy. This method tries providing the best possible initial weights from samples of the targeted data. Two examples of such methodology are Principal Component Analysis (PCA) [141], and Linear Discriminant Analysis (LDA) [5]. Both techniques were compared to random-weights initialization method and proved to be faster in terms of model convergence and stable in terms of model performance accuracy.

In summary, the learning-based methods have shown good performance in addressing various document layouts, including the complex ones. However, they are still suffering from some shortcomings. For instance, machine learning methods require more investigations to address data imbalance, developing representative features and derive accurate and automatic region-based segmentation. Despite feature generation using deep-learning methods, they are still requiring post-processing methods such as clustering or morphological cleaning to improve the segmentation outcomes.

4.2.4 Voronoi-Based Analysis. Segmentation of arbitrary document-layout is a challenging task. Arbitrary layouts have no specific shapes in general, but can be surrounded with polygonal shapes. Fortunately, Voronoi diagram is a solution that can define boundary points around arbitrary regions. It makes no assumptions about a document layout shape and can describe border points of various layouts as Kise's method [81, 83]. In Kise's method, the Voronoi diagram is constructed using connected components. Moreover, the analysis is delivered based on the selection of the Voronoi edges that are characterized by two features; distance, and area ratio. A drawback of this method is using the centroids of the connected components to define the Voronoi points. This is because connected components are non-convex in general, which makes a single-point representation inappropriate. In contrast, the two-points representation of each connected component was suggested in [2], [35], and [96]. Both algorithms derive a neighborhood graph from the area Voronoi diagram, where each node indicates a document element.

In the Voronoi analysis, every node may have many neighbors that can be identified by Voronoi edges. The existence of multiple Voronoi neighbors could lead to inaccurate region extraction. It appears vividly when a document image contains multi-sized text components. Consequently, document regions segmentation become imprecise [96]. Agrawal et al. [1] proposed an integration of Docstrum algorithm and Voronoi algorithm to track neighbor components and produce better segmentation.

Among the concerns of Voronoi analysis is the construction time of its diagram. It takes a considerable amount of time to generate a Voronoi diagram. First, it reduces connected components into dots. Then, it has two passes to generate Voronoi points of these document dots: (1) Voronoi points definition and (2) deleting all self Voronoi edges of each Voronoi point. These two passes of the Voronoi algorithm are computationally expensive especially for high-resolution document images [174].

4.2.5 Delaunay Triangulation Analysis. In general, Delaunay triangulation is a dual of Voronoi diagram, however, their edges are defined within document elements rather than between them [56]. Moreover, Delaunay edge-points simplify region segmentation rules: (1) the smallest edge-points represent text components on the same text line, (2) the largest edge-points represent text components between contiguous text lines, and (3) triangles that have sides larger than some

Fig. 10. An example of energy map text line segmentation as in [131].

pre-computed thresholds represent text column regions or margin borders [56]. Following these rules, the Delaunay triangulation was employed successfully to address text line segmentation in [169] and extracting authors' names and titles [170].

Other methods that are not within the above categories follow a bottom-up strategy but are not within the above categories. For instance, a morphological analysis approach manipulates document image pixels using a set of dilation and erosion operations to target document regions segmentation [33, 161]. Both studies targeted text versus non-text region extraction using shape structure of text features. Another study by Bukhari et al. [32] proposed active-contours-based curvy text-line segmentation. The method finds the ridges in central parts of text-lines. Then, it applies active-contours for text-lines extraction.

4.3 Top-Down Strategy

In this section, the discussion is covering four top-down categories; Texture-based Analysis, Run Length Smearing Algorithm (RLSA), DLA projection-profile, and White space analysis.

4.3.1 Texture-Based Analysis. Jain and Zhong [75] proposed a mask-based texture analysis to locate text regions written in different languages. The method was tested using English and Chinese documents and found useful even if the analyzed document contains text written in different languages. Another method assumes that the textual regions are darker than the background regions. In this case, an energy map can be generated to produce low responses for either pure foreground or background areas, while it yields high responses at edge text borders. For instance, Saabni and El-Sana [132] suggested a method that builds seem lines among text-lines using energy maps (see Figure 10). Saabni and El Sana's method requires repeated calculations of the global energy map to estimate local seam-lines. An improved version of Saabni and El Sana's method that avoids global re-computation of the energy map is proposed in [20]. This algorithm updates the energy map locally during text line detection. A detailed description and comparison of both algorithms are reported in [131]. A similar method analyzes the projection profile of the energy maps to enhance the overall text-line segmentation in historical manuscripts [17].

Image filtering can reveal strong document-image characteristics that are utilized in DLA. Six anisotropic Laplacian of Gaussian (LoG) filters and one isotropic Gaussian were used for document layout analysis in [48]. The response space is analyzed to find maximum peaks along specific orientations to detect text regions. Then, text-lines were extracted using K-means clustering based on orientation features. Similarly, another texture-based algorithm that coarsely locates main text regions using Gabor filter is described in [19].

Multiscale texture analysis may reveal useful information about a document layout at different scales. This characteristic of multiscale analysis can help in the analysis of degraded documents. For instance, a multiscale analysis may allow tracing of high responses at each upscale and map them back to its original level to extract document regions, as in [49]. The regions with back-traces may end-up detecting degraded text lines. Similar studies that analyze the behavior of multiscale texture analysis are reported in [18] and [19]. In [19], document images were filtered using Gabor filter to located different regions. Then, a minimization energy function was used to extract these

regions. This technique is extended in [18], where Gabor filter was applied at different angles spanning the interval [0°, 180°] to detect curvy regions that was detected in [19].

4.3.2 Run Length Smearing Algorithm (RLSA). Run Length Smearing Algorithm converts image-background to image-foreground if the number of background pixels between any two consecutive foreground pixels is less than a predefined threshold (i.e., it smears foreground pixels). RLSA was first introduced by Wahl et al. in [162] to conduct text-line analysis from the structure of simple document layouts. Shi and Govindaraju [147] proposed a modified RLSA to perform smearing on horizontal and vertical scans on two directions. Unfortunately, RSLA is very sensitive to writing styles such as multi-sized or curvy text [109, 147].

The multi-sized text issue is addressed using adaptive RLSA that updates its thresholds based on the algorithm local data characteristics [109]. This algorithm performs two runs of the RLSA on the connected components; first run is conducted to remove noisy obstacles, and the second run is performed to detect text lines. Another RLSA algorithm is designed to extract text lines using a generalized adaptive local connectivity map (ALCM) [156]. However, ALCM is sensitive to text heights, which may lead to a false text line extraction.

The RLSA can perform better, giving that some text structure prepossessing was conducted previously [4]. In general, RLSA is a robust and simple to apply technique, but it requires careful estimation of the algorithm thresholds. Direct application of RLSA on handwritten documents may result in low performance due to write-style heterogeneity [144].

4.3.3 DLA Projection Profile. Document projection profile method can be used to detect document regions [142]. Nagy et al. [105] proposed a X-Y cut algorithm that used projection profile to determine document blocks cuts. Usually, the X-Y cut algorithm suits structured document layouts that have fixed text regions and line spacing. Moreover, it depends heavily on clean document images without document border-noise to achieve proper analysis performance [142].

There are several modifications that were suggested to improve the performance of the original X-Y cut algorithm such as heuristic-based method [40]. Moreover, it was extended to find cuts based on projections of bounding-boxes [74]. Furthermore, another modification to the X-Y cut algorithm analyzes text regions using edit-cost evaluation metrics to guide segmentation decisions [157].

4.3.4 Whitespace Analysis. It is used to detect regions that can be isolated by spaces (i.e., background) from all directions. It assumes that all foreground regions are separated from each other by some whitespace. It can be formulated as a maximization optimization problem to find the maximal white rectangles in each direction [10, 126, 144]. The whitespace analysis suites the segmentation of regular document layouts [28]. This algorithm detects globally column text regions and conducts coarse analysis to extract column-text lines. Whitespace analysis is integrated with the X-Y cut algorithm in [85]. It finds proper cuts of the whitespaces to form homogeneous regions. In summary, the whitespace analysis methods suit structured documents with clear whitespaces separation among their regions.

4.4 Hybrid Strategy

The hybrid strategy is the integration of the bottom-up and top-down strategies. Even though the research in bottom-up and top-down algorithms are well established, there are still many challenging issues that neither bottom-up nor top-down algorithms can address appropriately.

Usually, a design of DLA technique requires the analysis objectives such as regional detection, text-lines extraction, and so on. In addition, each analysis objective needs analysis parameters such

as font size, average text-line gaps, or average word gaps. These parameters could be estimated using one analysis strategy and document segmentation can be achieved using another strategy.

Each strategy has its strengths and weaknesses. For instance, whitespace analysis (i.e., top-down strategy) can perform region segmentation faster than connected component analysis (i.e., bottom-up strategy) because the whitespace analysis focuses on the background data instead of the foreground (i.e., whitespaces). On the other hand, the whitespace analysis may lack segmentation precision especially if the document-regions have irregular layout structures. The connected-component analysis is better at extraction and detecting regions' elements. However, it may be difficult for the connected-component analysis to extract an element that has text-touching issues in multiple regions. Consequently, these highlighted positives and negatives of the whitespace and connected component analyses can be integrated to boost their segmentation decisions. For example, text-touching can be resolved if whitespace information along a region side is known [126] or using machine learning to estimate split-merge parameters of regions [168]. Another hybrid technique that integrates learning-based analysis, RLSA, and whitespace analysis is described in [25]. First, the algorithm detects text and non-text objects using neural networks. Then, it performs RLSA followed by whitespace analysis to determine region boundaries.

Unlike other hybrid approaches, rule-based heuristics using connected components can be considered as a hybrid technique. The technique starts with a top-down view that uses knowledge-based rules to determine text by detecting document elements with minimum lengths [84]. Then, it analyzes the connected components based on the context and geometric characteristics to group them into text lines. Similarly, Tran et al. [160] proposed a heuristic hybrid approach that detects text and non-text regions using minimum homogeneity algorithm.

In general, hybrid techniques may provide robust analysis and can deal with arbitrary or complex document layouts. Even though studies such as [25] and [75] claimed generalization of their algorithms to cover any document layouts, their experiments were conducted on a small population. Moreover, there are some algorithms that combine techniques of the same strategy such as the integration of two bottom-up algorithms in [1] and [159] or top-down algorithms in [85]. To sum up, the hybrid strategy is rarely investigated in comparison to bottom-up or top-down strategies. The integration of methods may reveal robust algorithms for complex document layout analysis. Therefore, more efforts are needed to study hybrid techniques in the future.

4.5 Post-Processing

Once a DLA method completes the analysis phase, some post-processing steps might be required to deliver the segmentation outcomes. In general, the post-processing phase is an optional in most DLA algorithms. There are four main reasons why previous methods have introduced a post-processing phase:

- It allows algorithms to reason performance degradation in particular cases.
- Post-processing could be conducted to give an initial proof for future directions.
- It could be used to show method scalability to cover various document layouts with some simple tweaks.
- It could be necessary to compensate for method segmentation limitations.

Usually, the first and last reasons above were the most frequent causes for introducing a post-processing phase in various DLA algorithms. For instance, Ramel et al. [126] suggested post-processing using human interaction to perfectly extract each region. Most of the deep-learning algorithms require either superpixel clustering or morphological cleaning [43, 70, 113, 123, 167].

Document layouts variability in some datasets requires post-processing to compensate for the algorithmic parameters such as extreme region shapes, and irregular spacing. Examples of such

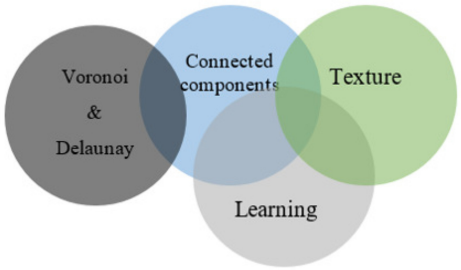


Fig. 11. Bottom-up techniques relationship set-view.

post-processing that uses context, spatial relationships of document elements to enhance the analysis outcomes are [31] and [132]. Another work that has fixed segmentation errors such as splitting connecting components of two text lines [131]. Furthermore, post-processing could be a mandatory task in some situations to compensate for the algorithm's speed in tradeoff segmentation accuracy. For example, texture-based algorithms that classify image-pixels into text or non-text can be considered fast algorithms. However, it may fail to determine region boundaries accurately due to the mixed characteristics of transition pixels [77, 103]. Most of the learning-based DLA methods require superpixel clustering or morphological cleaning to post-process the final results [43, 113, 123].

4.6 Discussion

The bottom-up analysis strategy is the most frequently used methodology in designing document layout analysis. Regardless of the required space and time complexity of the bottom-up methods, its positive characteristics, such as handling complex layouts, have attracted researchers to follow it. According to this review, it has been used in 61% of 79 studies. The literature sample is not exhaustive, however, we can infer that the bottom-up is the dominant strategy.

The basic building block of most of the bottom-up techniques is the connected components. Many studies have proposed techniques using them instead of pixels. They are used in almost all bottom-up techniques such as the learning-based [31], texture-based [125], SIFT [63], Voronoi [2], and Delaunay [56]. Figure 11 illustrates the relationship between these techniques.

The top-down methods may require less amount of data in comparison to bottom-up. For instance, the white space analysis locates spaces surrounding regions while machine-learning methods may need pixel-level information about every component, such as background, text, or non-text. Usually, the top-down strategy works perfectly on regular layouts, however, it requires a clean and skew corrected document images such as [28], [126], and [144]. Yet, the top-down strategy techniques are important for DLA because most contemporary documents have regular or Manhattan layouts.

Although hybrid document layout analysis tends to utilize the strengths of both essential strategies in one methodology, it is the least frequently used DLA methodology. Based on our review population of DLA research, around 6% of 79 studies have considered hybrid methodology. Therefore, it could be an opportunity for researchers to develop and study the integration of various DLA techniques.

Moreover, we note that recent methods that used deep-learning are combined clustering or cleaning methods to the pipeline of the layout analysis to deliver the final page segmentation [38, 70, 113]. The integration of methods in these studies is not considered as a hybrid methodology but as a post-processing procedure. Because the integrated methods are usually considered to perform minimal analysis.

The importance of the preprocessing phase including binarization, noise removal, and de-skewing cannot be neglected. In the literature, there are several studies that did not detail the preprocessing phase of their methods. They assume an input of preprocessed document-images. Others described their method requirements that include performing binarization or de-skewing on the input document images before DLA. The binarization is still an active research topic due to its importance in DLA pipeline [122].

Document language is another important concern in DLA. The analysis might be completely different from one language to another. For example, printed/typed English documents can be broken down into columns, paragraphs, text lines, words, connected components, or characters, while Arabic documents can be broken down to the level of part of an Arabic word (PAW). In Arabic language, it would be very challenging to analyze documents at the level of characters because of several reasons: (1) Arabic characters could have up to four shapes based on the character location [27], (2) it is written cursively in typed and handwritten styles, and (3) it could have irregular spacing among PAWs with letter elongation strokes. Therefore, documents with different languages may need to be treated differently in DLA.

Table 2 summarizes the literature review on document analysis. Most of the studies addressed contemporary documents. This is reasonable because modern life requires smart offices, libraries, and the like. Therefore, at this stage, it is not enough to have digitized documents but software and tools to facilitate such digitization for document understanding, retrieval, clustering, or recognition. In addition, there are few DLA studies on handwritten documents in comparison to printed documents. We note that the handwritten documents listed in Table 2 are either historical documents or mixed documents. Furthermore, most of the previous studies were conducted on English documents, which may be attributed to the availability of some benchmark datasets and/or the easy access to repositories of scientific journals which are mostly written in English. This explains further why most of the analyzed research either multi-column or Manhattan layouts 37% and 35%, respectively of the 79 DLA studies. Figure 12 illustrates the contribution of research per document category based on our non-exhaustive list of DLA population. The statistics in Figure 12 is affected by the selection of papers. To keep this article to a reasonable size, we mainly concentrated on Latin and Arabic languages. Representative articles of other languages are included for reference Figure 12(c).

5 DOCUMENT LAYOUT ANALYSIS EVALUATION

In general, the evaluation process of the DLA algorithms consists of two aspects; datasets and evaluation metrics. We discuss in this section the experimental settings and the performance evaluation metrics that have been used in document layout analysis.

5.1 Datasets

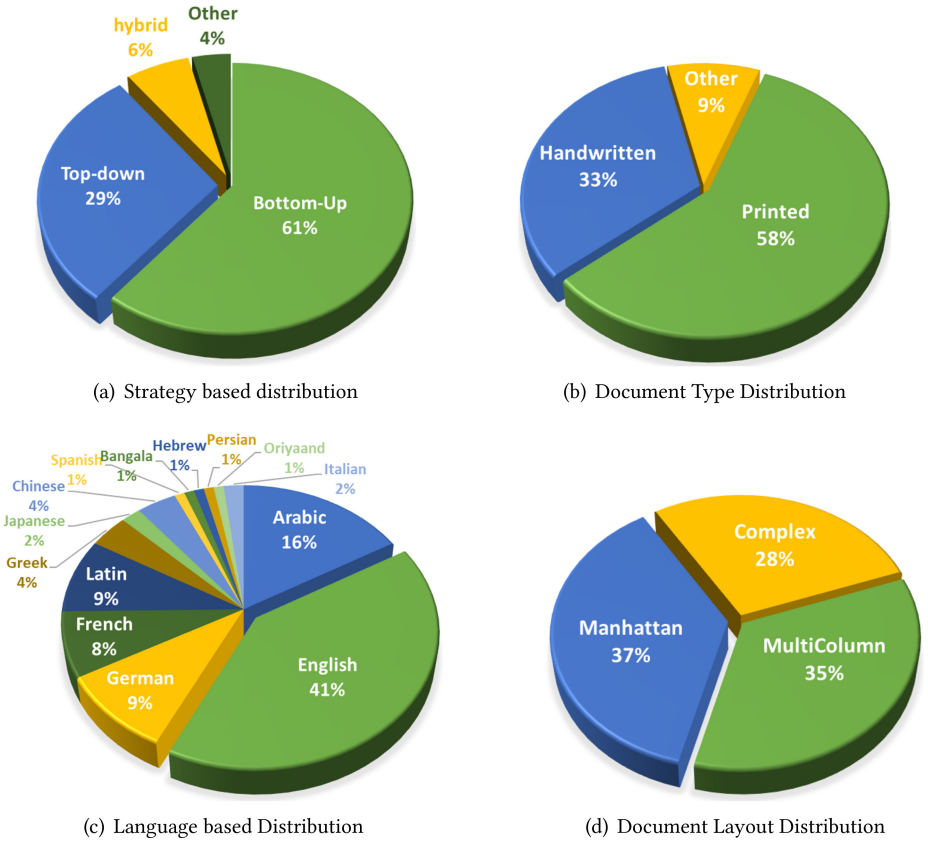
There are several datasets that can be used for document layout analysis. Table 3 illustrates a list of benchmark datasets and their statistics. Based on document types, there are three main categories for document datasets; printed, handwritten, and mixed (i.e., documents that contain typewritten and handwritten).

Printed datasets are generally developed to test the DLA methods that target contemporary documents understanding. An example of such datasets is developed by the University of Washington (UW-3) [146]. It consists of 1,600 skew corrected technical English articles. The UW-3 dataset provides bounding-box coordinates as ground-truths for both text, and non-text zones. Each region in a zone is labeled as text, math, table, or figure. It is suitable for page layout analysis of technical articles that targets text versus non-text extraction. Another common dataset is published by Pattern Recognition & Image Analysis (PRImA) Research Lab [13]. It is designed for evaluating

Table 2. Document Layout Analysis Algorithms Summary

Ref	STGY	Mthd	Documents			RST	Ref	STGY	Mthd	Documents			RST
			TYP	Lang.	LYT					TYP	Lang.	LYT	
[22]		ML	P	Multi	MC	Tline	[83]		Vorni	P	Multi	MC	CLN
[52]		ML	P	Eng	MB	Txt-z	[35]		Vorni	P	Eng	MB	LDw
[62]		ML	P	Eng	MB	DCEl	[2]		Vorni	MX	Multi	MB	Txt-z
[164]		ML	MX	Multi	MC	MO	[1]		Vorni	MX	Multi	MB	Txt-z
[45]		ML	MX	Multi	MC	Txt-z	[174]	ST1	Vorni	P	Eng	MC	Txt-z
[163]		ML	MX	Multi	MC	MO	[169]		Dlny	P	Eng	MC	Txt-z
[123]		ML	H	Eng	MB	MO	[170]		Dlny	P	Eng	MC	MO
[71]		ML	P	Arb	CPX	MO	[56]		Dlny	H	Multi	CPX	Page
[38]		ML	P	Multi	MB	MO	[142]		PP	P	Eng	MC	Txt-z
[42]		ML	H	Multi	CPX	MO	[109]		RLSA	P	Multi	MC	MO
[63]		ML	P	Grn	MC	Tline	[144]		WSpce	P	Eng	MC	Tline
[65]		ML	P	Eng	MB	Txt-z	[28]		WSpce	P	Eng	MC	MO
[44]		ML	P	Multi	MB	MO	[75]		Txtr	P	Multi	MC	Txt-z
[103]		ML	MX	French	MB	Txt-z	[74]		PP	P	Eng	MC	Txt-z
[165]		ML	MX	Multi	CPX	MO	[157]		PP	P	Eng	MC	CLN
[30]		ML	P	Eng	CPX	Txt-z	[105]		PP	P	Eng	MB	MO
[31]		ML	H	Arb	CPX	Txt-z	[162]		RLSA	P	Eng	MC	Txt-z
[167]	ST1	ML	H	Multi	CPX	MO	[49]		Txtr	MX	Multi	MC	Tline
[70]		ML	H	Eng	MB	Tline	[156]	ST2	RLSA	H	Multi	MB	Tline
[43]		ML	H	Multi	CPX	Txt-z	[131]		Txtr	H	Multi	MC	Tline
[141]		ML	P	Multi	CPX	MO	[17]		Txtr	H	Multi	MB	Tline
[113]		ML	H	Eng	MB	MO	[19]		Txtr	H	Arb	CPX	Txt-z
[5]		ML	P	Multi	CPX	MO	[48]		Txtr	H	Arb	CPX	Txt-z
[59]		Txtr	P	Eng	MC	Tline	[132]		Txtr	H	Multi	MB	Tline
[101]		Txtr	P	French	MB	MO	[20]		Txtr	H	Multi	MB	Tline
[64]		Txtr	P	Eng	MB	Txt-z	[4]		Txtr	H	Multi	MB	Tline
[77]		Txtr	P	Eng	MC	MO	[148]		Txtr	H	Arb	MB	Tline
[76]		Txtr	P	French	MC	Txt-z	[147]		RSLA	H	Eng	CPX	Tline
[102]		Txtr	P	Multi	CPX	Txt-z	[18]		Txtr	H	Arb	CPX	Txt-z
[159]		CC	P	Eng	CPX	Txt-z	[160]		HSTC	P	Eng	MC	MO
[125]		CC	H	Hb	CPX	Tline	[126]	ST3	WSpce	P	NA	MB	Txt-z
[111]		CC	P	Eng	MC	Txt-z	[25]		ML	MX	Multi	MB	Txt-z
[152]		CC	P	Eng	MC	Txt-z	[84]		HSTC	H	Eng	MB	Tline
[96]		Vorni	P	Eng	MC	Word	[161]		Mor	P	Eng	MC	Txt-z
[81]		Vorni	P	Multi	MC	Txt-z	[33]	Other	Mor	P	Eng	MC	Txt-z
							[32]		Snakes	H	Multi	MB	Tline

– STY: Strategy, ST1: Strategy 1 (Bottom-UP), ST2: Strategy 2(Top-Down), ST3: Strategy 3(Hybrid), Ref: Reference, Mthd: Method, TYP: Type, Lang: Language, LYT: Layout, RST: Result ML: Machine Learning, PP: Projection Profile Analysis, CC: Connected Component Analysis, Mor: Morphological. Analysis, HSTC: Heuristics, WSpace: Whitespace, Txtr: Texture, Vorni: Voronoi, P: Printed, MX: Mixed Documents, H: Handwritten, CPX: Complex, MC: Multi-Column, MB: Manhattan-based, Arb: Arabic, Eng. English, Fr. French, Hb: Hebrew, Grn: German, Multi: Multiple languages, Txt-z: Text zones (includes paragraphs, text-blocks, and other blocks), TLNE: Text line, L-D: Line drawing, DCEL: decorative elements, MO: Multi-objective (such as periphery, background, text block, and decoration), T-A: Title and Author extraction, CLN: Column regions.



(a) Strategy based distribution

(b) Document Type Distribution

(c) Language based Distribution

(d) Document Layout Distribution

Fig. 12. Document layout analysis techniques statistics.

Table 3. Document Layout Analysis Datasets

Datasets	Age	Data Statistics				Ground Truth			Type	ICDAR	Refs & Year
		Wrts	Typ	Doc.	Lang.	TL	CC	BB			
OHG	HS	1	H	596	Eng.	Yes	Yes	Yes	LA	NA	[123], 2018
Diva-hisdb	HS	NA	H	150	Multi	Yes	Yes	Yes	LA	ICDAR17	[151], 2017
Parzival	HS	3	H	47	Gmn	Yes	No	No	LA/SP	NA	[78], 2009
GW20	HS	1	H	20	Eng.	Yes	Yes	Yes	LA/SP	NA	[86], 2007
Saint Gall	HS	1	H	60	Latin	Yes	No	No	LA/SP	NA	[78], 2006
IMPACT	HS	NA	P	7K	Multi	No	No	Yes	LA	ICDAR*	[36], 2000
BCE-Arabic-v1	CN	NA	P	1833	Arb.	No	No	Yes	LA	NA	[133], 2016
UW-3	CN	NA	P	1600	Eng.	Yes	Yes	Yes	LA	NA	[146], 2013
MAURDOR	CN	NA	MX	2.5K	Multi	Yes	Yes	Yes	LA/SP	NA	[25], 2013
CENIP-UCCP	CN	200	H	400	Urdu	Yes	No	No	LA	NA	[129], 2012
LAMP	CN	NA	MX	203	Multi	No	No	Yes	LA	NA	[176], 2010
PRImA	CN	NA	P	305	Eng.	Yes	Yes	Yes	LA/SP	ICDAR+	[13], 2009

– CN: Contemporary, HS: Historical, P:Printed, H: Handwritten, LA: Layout Analysis, Sp: Spotting NA: Not Applicable, Multi: different languages, Pub: Publishing Year, CNU: Ho Chi Minh National University, TL: Text Lines, CC: Connected components or words, BB: Block or region; ICDAR+: includes Page Layout competitions (2001, 2003, 2005, 2007, 2009, 2011, and 2015), ICDAR*: includes Page Layout competitions (2013, and 2015).

methods that target modern document layout. PRImA dataset consists of 305 pages from various sources with emphasis on technical publications and magazines. Page Analysis and Ground-truth Elements (PAGE) format was used to prepare the PRImA ground truth [119]. Recently, a collaboration between Boston University, Cairo University, and Electronics Research Institute have yield a BCE-Arabic dataset [133]. The BCE-Arabic dataset consists of 1,833 printed pages collected from 180 books. The ground-truth is generated manually using several tools such as Pixlabeler [137], Groundtruthing Environment for Document Images (GEDI) [54], and "Document, Image, and Video Analysis, Document Image Analysis" (DIVADIA) [78]. The dataset suites various analysis objectives such as text only analysis (1,235 pages), text vs. images (383 pages), text vs. graphic elements (179 pages), text vs. tables (24 pages), single or double column text vs. images (29 pages). Similarly, a large dataset is developed under the project of IMProving ACcess to Text (IMPACT) [36]. It contains 70,000 historical printed pages of 17th–20th centuries. The data were collected from various sources in 17 languages.

There are few handwritten-document datasets that are published for layout analysis. For instance, the famous George Washington papers (GW20) dataset consists of 20 pages segmented into text lines, words, and word classes [86, 128]. Saint Gall medieval manuscripts are written in Latin by a single writer [60]. It consists of 60 pages in total divided into 20 pages for training, 30 pages for testing, and 10 are selected for validation. The Saint Gall dataset is mainly used for text line analysis and keyword spotting. Moreover, Parzival dataset represents the epic poem Parzival by Wolfram Von Eschenbach [60]. It consists of 47 pages written in German in the $13th$ century by three writers. It is divided into 24 pages for training, 14 pages for testing, and two pages for validation. The ground truth of both datasets (Saint Gall and Parzival) are extracted using DIVADIA software. Another handwritten dataset called CENIP-UCCP was collected by the Center of Image Processing-Urdu Corpus Construction Project (CENIP-UCCP) [129]. It contains 400 text pages written by 200 writers in Urdu. The ground-truth is provided at the text-lines level. Recently, Lorenzo et al. [123, 124] published a large dataset of the 18 century Spanish notarial deeds, the office of mortgage register (OHG). The OHG consists of 596 historical documents where each document is associated with a PAGE format ground truth.

Even though analyzing documents that contain mixed text i.e., handwritten and printed content is seldom in the literature, there are few examples. The Laboratory for Language and Media Processing (LAMP) at the University of Maryland developed a mixed dataset [176]. It contains 203 pages; 109 in Arabic and 94 in English. It could be used for analyzing complex document layouts for text, image, and signature extraction.

Several other datasets can be found within the ICDAR and ICFHR Page Segmentation Competitions [8, 15, 123, 124, 151], and and MAURDOR campaigns [50]. Usually, these datasets are associated with software tools for evaluation purposes.

5.2 DLA Evaluation Metrics

The performance evaluation of document layout analysis methods is a critical task. It remained subjective for several years in the past [35, 75, 83]. The subjective performance evaluation does not allow method comparison, and hence hinders DLA progress. Apart from the subjective evaluation, there are several DLA methods that concentrate on benchmarking using simple measures such as precision and recall [103, 131, 143, 160]. Moreover, element counting is another example of a simple evaluation metric that uses the count of correctly segmented connected components as in [148], words [96], or text-lines as in [20], [63], and [144]. This evaluation method is also named as recognition rate as in [30], [33], and [131]. All these metrics are based on pixel-level evaluation. Although these simple measures are objective and allow method comparisons, they miss being customizable.

The customizable performance evaluation methods allow DLA algorithms to be tested on several applications (i.e., analysis objectives) such as extracting paragraphs, text-lines, words, tables, figures, and the like. Usually, such methods use more sophisticated metrics. The ICDAR competition on Page Segmentation series has evolved its evaluation metrics from element counting-based [9] to more customizable metrics [8, 11, 14, 15, 151].

5.2.1 Pixel-Level Evaluation Framework. Pixel-level Evaluation Framework (PLEF) metric is considered in several studies and competitions [9, 47, 67, 117, 118]. In the pixel-level evaluation, the performance evaluation is based on counting the number of pixels matches between the segmentation results and their corresponding ground-truth. Moreover, these methods can use solely foreground pixels to the compute the matching score of document-image I [167]. The segmentation results and ground-truth elements are matched using global match table $MS(i, j)$ as in the following equation [9, 69]:

$$MS(i,j) = \alpha \frac{(T(G_j \cap R_j \cap I))}{(T((G_j \cup R_j) \cap I))} \quad \text{where} \quad \alpha = \begin{cases} 1, & \text{if} \quad g_i = r_i \\ 0, & otherwise \end{cases} \tag{4}$$

where $T(.)$ is a function to count pixel matches of all ground-truth set elements G_j of region j to all segmented set elements R_i of the segmented region i. The g_j and r_i are entities of j ground-truth and i segmented regions, respectively.

Finally, the matching table is used to compute the DLA performance as $F_{measure}$ using detection rate (DR) $\approx$ Pr and recognition rate (RR) $\approx R$, which are computed using Equations (5) and (6) [9, 69, 93]:

$$DR = w_1 \frac{one2one}{N_i} + w_2 \frac{one2many}{N_i} + w_3 \frac{many2one}{N_i} \tag{5}$$

$$RR = w_4 \frac{one2one}{M_i} + w_5 \frac{one2many}{M_i} + w_6 \frac{many2one}{M_i} \tag{6}$$

where N_i and M_i are the ground-truth and segmented elements of entity i, respectively. *one2one*, *one2many*, and *many2one* are computed from the matching table $MS(i, j)$, and w_1, w_2, w_3, w_4, w_5 and w_6 are predefined weights that can be set to evaluate a particular analysis objective. Finally, the results of the above equations can be combined to report the $F_{measure}$ as follows:

$$F_{measure} = \frac{2 \times DR \times RR}{DR + RR} \tag{7}$$

5.2.2 Region-Level Evaluation Framework. The Region-level Evaluation Framework (REF) is a type of DLA method performance evaluation that concentrates on abstract level evaluation. In other words, segmentation errors can be categorized as major and minor [161]. For example, consider the segmentation error that happens because of text-touching. It shows two text-lines in the same text-column as one text-line. This error is a minor error and can be ignored if the DLA application is text-column extraction. The same error is treated as a major error if the DLA application is paragraph extraction.

The Intersection over Union (IoU) and the frequency weighted (f.w.IoU) are two examples of region-based evaluation [43, 167]. This method was borrowed from computer vision for performance evaluation of object detection methods. In computer vision, algorithms detect objects by surrounding them with bounding boxes. Then, the quality of the object detection is computed using the Intersection over Union between the ground-truth object and the detected object bounding boxes. A similar scenario is mapped in document layout analysis where objects are extracted from document regions. This metric reports the average of the intersection of two regions divided by

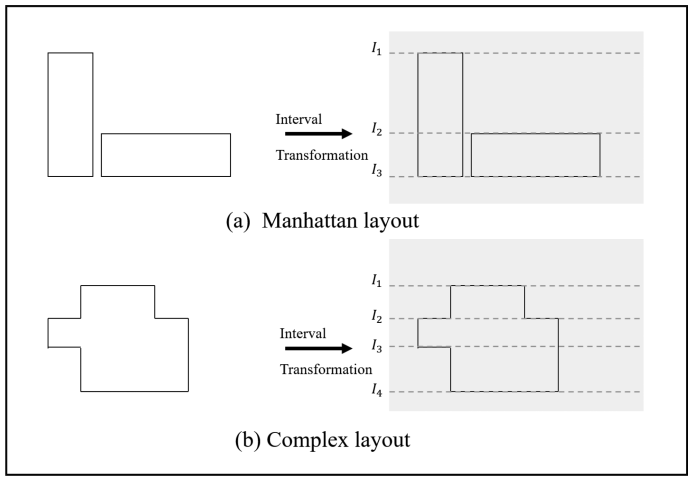


Fig. 13. Region transformation example.

their union for all segmented regions. The average Intersection over Union is computed as follows:

$$IoU = \alpha_1 \frac{G_i \cap R_i}{t_i + (G_i \cup R_i) - (G_i \cap R_i)} \tag{8}$$

where $\alpha_1 = 1/N$, and N is the number of classes, t_i is the total number of pixels in class i, G_i and R_i are the ground-truth and segmented regions, respectively. The frequency weighted IoU is computed as follows:

$$f.w.IoU = \alpha_2 \frac{t_i \times (G_i \cap R_i)}{t_i + (G_i \cup R_i) - (G_i \cap R_i)} \tag{9}$$

where $\alpha_2 = 1/t_{i_k}$, and t_{i_k} is the number of elements in class i. The intersection over union metric could provide more tolerance to segmentation minor errors. However, it may produce different scoring results for the correct segmented result produced by two document analysis algorithms [167]. The IoU metric considers complete elements of the intersection. Instead, Wick and Puppe [167] suggested the use of foreground elements to calculate IoU.

5.2.3 Customizable Evaluation Framework. In contrast to PEF and REF, the Customizable Evaluation Framework (CEF) is designed to give an in-depth performance evaluation of layout analysis. It was first introduced in 2007 by Antonacopoulos and Bridson [12]. Since then, it is used in the performance evaluation of ICDAR Page Segmentation Competitions series [8, 11, 14, 15, 43].

The CEF performs three main steps: (1) region transformation, (2) region correspondence, and (3) error qualification and quantification. The region transformation is an essential step that allows the CEF to address the evaluation of complex layouts more efficiently than PEF or REF. In this step, both regions of the ground-truth and segmented results should be transformed into interval representation. In interval representation, a single interval is a maximal rectangle that fits horizontally inside a region with a width that starts from a given point on a vertical edge and ends at the lowest possible point on the vertical edge in the opposite direction. Based on region layouts, the transformation may produce a single interval for Manhattan layouts, or $N = 2, 3, 4, \ldots n$ intervals for complex layouts. Figure 13 shows an example of the region to interval transformation for Manhattan and complex regions into three and four intervals, respectively.

Once all regions of both ground-truth and segmentation results are transformed into interval representation, the framework determines region correspondence by combining them. In the

combined interval representation, the interval lines are examined to detect major overlaps for correspondence definition. Consequently, three states of overlapping are defined:

- Segmentation interval over nothing
- Segmentation interval over a ground-truth interval
- Nothing over Ground-truth interval

Given these three-correspondence states, the analysis quality can be easily captured in terms of segmentation errors:

- *Merge:* Occurs if one segmented result has multiple correspondences to two or more ground-truth regions.
- *Split:* Occurs if two or more segmented regions have correspondence to only one ground-truth region.
- *Miss:* A ground-truth region established no correspondence with any segmented results.
- *Partial Miss (PMiss):* Occurs if a segmented region overlaps partially a ground-truth region.
- *False Detection:* Occurs if a segmented region established no correspondence with any ground-truth region.

The segmentation errors can have different significance depending on the document context and analysis application. For a context-based example, a merger between paragraphs within a single column is less significant than a merger of paragraphs between two columns. On the other hand, the OCR application may tolerate a merger of graphical regions but not text regions. During the region correspondence, the actual and the affected areas of these overlaps are computed. The affected areas belong to non-overlap parts (i.e., segmentation actual error area). To compute the success rate of the analysis based on all types of errors, first the error rate ER_i is computed by multiplying the affected area of error i by a predefined error weight α_i. The value of α_i depends on the context and application of the analysis. The affected area is either the segmentation result area in the case of false detection or the count of the foreground pixels in the case of other error types. Then, the final error weight w_i is computed as follows:

$$w_i = \frac{(N-1)ER_i + 1}{N} \tag{10}$$

where $1 \leq N \geq 5$ is the number of error types that is considered in the evaluation scenario. The success rate SR is computed as follows:

$$SR = \frac{\sum_{i=1}^{N} w_i}{\sum_{i=1}^{N} \frac{w_i}{1-ER_i}} \tag{11}$$

In summary, the pixel-based evaluation metric is an aggressive and rigid method to evaluate the DLA performance. In addition, it does not allow in-depth segmentation error analysis. On the other hand, it can be used for benchmarking data or method performance. Second, pure region-based evaluation methods are rarely used. They are mainly borrowed from the Computer Vision field. This evaluation metric may suit evaluating methods that address large region analysis such as text-columns, figures, tables, logos extraction. Third, the customizable evaluation framework gives an in-depth segmentation error analysis. Moreover, it supports various document layout ranges from Manhattan to complex ones. In addition, several evaluation scenarios can be customized by changing error significance using the local weight α_i. Finally, there are several other performance evaluation metrics that are borrowed from other fields to evaluation DLA methods. For instance, the Precision and Recall that are used in information retrieval field. Table 4 lists examples of DLA quantitative results.

Table 4. Examples of DLA Evaluation Metrics & Results

Metric	Ref	Database	Results (%)	Metric	Ref	Database	Results (%)
Success Rate	[43], 2017	Multi	98	Sub.	[111], 1993	Private	NA
	[167], 2018	Multi	98.4		[105], 1984	NA	NA
	[161], 2017	Multi	89.7		[162], 1982	Private	NA
	[103], 2015	IMPACT	96.00	$F_{Measure}$	[70], 2018	cBAD	96.7
	[44], 2015	Multi	97		[102], 2017	PRImA	71
	[56], 2015	Multi	100		[160], 2016	Multi	94.58
	[59], 2014	Parzival	96.80		[159], 2015	PRImA	96.80
	[45], 2014	Multi	97.90		[42], 2016	DIVA-HisDB	90
	[163], 2013	Multi	97.47		[103], 2015	IMPACT	75.00
	[22], 2013	Multi	96.3		[18], 2015	Private	90.0
	[63], 2012	Saint Gall	98.65		[156], 2015	Multi	98.66
	[4], 2011	Multi	98.55		[131], 2014	Private	98.00
	[20], 2011	Private	98.00		[17], 2014	Private	99.97
	[132], 2011	ICDAR2007	98.90		[19], 2014	Private	98.84
	[4], 2011	Private	76.00		[49], 2014	Multi	99.69
	[52], 2011	PRImA	94.50		[48], 2013	Private-IHP	92.95
	[33], 2011	Multi	99.35		[125], 2013	Multi	91.42
	[30], 2010	Multi	95.72		[31], 2012	Private	94.68
	[62], 2010	Private-Psalter	91.35		[65], 2011	Private	91.4
	[148], 2009	MADCAT	99.50		[64], 2010	Private-Psalter	96.80
	[144], 2008	ICDAR2007	98.40		[109], 2010	Private	84.80
	[77], 2008	Private	87.50		[32], 2009	Multi	96.30
	[71], 2004	Private	98.43		[126], 2007	Private	86.30
	[96], 2004	Multi	99.05	IoU	[38], 2018	Multi	93
Sub.	[147], 2004	USPS	93.00		[113], 2018	DIVA-HisDB	83.39
	[84], 2008	Private	NA		[5], 2017	DIVA-HisDB	93.39
	[174], 2005	UW-1	NA	Other	[141], 2017	DIVA-HisDB	NA
	[170], 2004	Private	NA		[164], 2014	Multi	5.49
	[169], 2003	Private	NA		[25], 2014	MAURDOR	57.80
	[28], 2003	UW-3	NA		[165], 2014	Multi	6.46(eer)
	[81], 1998	Multi	NA		[101], 2013	Private	74.00
	[83], 1997	Private	NA		[142], 2011	UW-3	7.50
	[152], 1997	Private	NA		[2], 2010	Multi	70.78
	[75], 1996	Private	NA		[1], 2009	UW-3	78.30
	[35], 1995	Private	NA		[76], 2005	Private	91.50
	[157], 1995	Private	NA		[74], 1995	UW-1	0.37

– Sub: Subjective, Other: such as Error Rates, Detection Rate, Precision, Recall, Jaccard index.
– Multi: Multiple data sets such as ICDAR Page Segmentation Competitions 2007, 2009, 2013, 2015 & 2017, Saint Gall, Parzival, UW, cBAD, GW, PRImA, CNU, or Private; IHP: Islamic Heritage Project, UW:University of Washington.
– ICDAR2007: Handwritten Segmentation Contest dataset, ICDAR2009: page segmentation competition.

6 CONCLUSIONS

This survey presents a thorough overview of the document layout analysis methods. The general DLA framework includes three main tasks; document preprocessing, analysis (i.e., page segmentation), and performance evaluation. Although the preprocessing is an important step in the pipeline of DLA, it has been neglected in several DAL methods. For example, the deep-learning methods

ignore document binarization and use complete pixel intensities to generate features of various document structures. In this sense, the document binarization is discouraged because it drops massive image information that is utilized by learning-based methods. On the other hand, document binarization is still an active research topic and a challenging task for historical manuscripts. For example, in DIBCO 2017 Binarization Contest series, 26 binarization methods were submitted for evaluation by 18 research groups. The DIBCO 2017 announced that the machine-learning binarization methods were the best in the contest. Another recent example is DeepOtsu method. In general, the DLA preprocessing phase includes three sub-tasks (viz. document skew correction, image enhancement, and binarization).

Second, document layout analysis is divided into three phases; initial analysis, segmentation, and post-analysis. The initial analysis helps in understanding the document basic structures such as word-spacing, font sizes, and the like. Usually, these parameters are used in the segmentation phase.

The bottom-up analysis methodology is used intensively by researchers due to its ability to handle the document analysis at fine levels. In addition, it suits the analysis of a plethora of document layouts. In contrast, bottom-up analysis requires a longer analysis time than top-down approaches. For example, a complex document layout can be analyzed using a top-down approach in 0.7 seconds with 78.5% success rate [161]. Therefore, an integration of both methodologies may yield faster and accurate DLA algorithms.

Currently, deep-learning DLA methods are producing cutting-edge results in the DLA field. Usually, such methods require long training time and huge data, however, they show high capabilities to cope with a wide range of document classes. In addition, these methods may require some post-processing (i.e., fine-tuning) to adjust their final analysis outcomes. Therefore, more efforts are needed to study and advance such technology towards developing multi-class document layout analysis.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Mudit Agrawal and David Doermann. 2009. Voronoi++: A dynamic page segmentation approach based on Voronoi and Docstrum features. In *The International Conference on Document Analysis and Recognition*. IEEE, 1011–1015.

[2] Mudit Agrawal and David Doermann. 2010. Context-aware and content-based dynamic Voronoi page segmentation. In *The 8th IAPR International Workshop on Document Analysis Systems*. ACM Press, New York, 73–80.

[3] Prakash K. Aithal, G. Rajesh, Dinesh U. Acharya, and P. C. Siddalingaswamy. 2013. A fast and novel skew estimation approach using radon transform. *International Journal of Computer Information Systems and Industrial Management Applications* 5 (2013), 337–344.

[4] Alireza Alaei, Umapada Pal, and P. Nagabhushan. 2011. A new scheme for unconstrained handwritten text-line segmentation. *Pattern Recognition* 44, 4 (2011), 917–928.

[5] Michele Alberti, Mathias Seuret, Vinaychandran Pondenkandath, Rolf Ingold, and Marcus Liwicki. 2017. Historical document image segmentation with LDA-initialized deep neural networks. In *The 4th International Workshop on Historical Document Imaging and Processing*. ACM, 95–100.

[6] Adnan Amin and Sue Wu. 2005. A robust system for thresholding and skew detection in mixed text/graphics documents. *International Journal of Image and Graphics* 5, 2 (Apr. 2005), 247–265.

[7] Khalid M. Amin, Mohamed Abd Elfattah, Aboul Ella Hassanien, and Gerald Schaefer. 2014. A binarization algorithm for historical Arabic manuscript images using a neutrosophic approach. In *The 9th International Conference on Computer Engineering & Systems*. IEEE, 266–270.

[8] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. 2011. Historical document layout analysis competition. In *International Conference on Document Analysis and Recognition*. IEEE, 1516–1520.

[9] A. Antonacopoulos, B. Gatos, and D. Karatzas. 2003. ICDAR 2003 page segmentation competition. In *The 7th International Conference on Document Analysis and Recognition*. 688–692.

[10] A. Antonacopoulos and R. T. Ritchings. 1995. Representation and classification of complex-shaped printed regions using white tiles. In *The 3rd International Conference on Document Analysis and Recognition*, Vol. 2. IEEE Comput. Soc. Press, 1132–1135.
[11] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. 2009. ICDAR2009 page segmentation competition. In *The 10th International Conference on Document Analysis and Recognition*. 1370–1374.
[12] Apostolos Antonacopoulos and David Bridson. 2007. Performance analysis framework for layout analysis methods. In *The 9th International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 2. IEEE, 1258–1262.
[13] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. 2009. A realistic dataset for performance evaluation of document layout analysis. In *The 10th International Conference on Document Analysis and Recognition*. IEEE, 296–300. DOI: https://doi.org/10.1109/ICDAR.2009.271
[14] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. 2013. ICDAR2013 competition on historical newspaper layout analysis (HNLA'13). In *The 12th International Conference on Document Analysis and Recognition*. IEEE, 1454–1458.
[15] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. 2015. ICDAR2015 competition on recognition of documents with complex layouts. In *The 13th International Conference on Document Analysis and Recognition*. IEEE, 1151–1155.
[16] Manivannan Arivazhagan, Harish Srinivasan, and Sargur Srihari. 2007. A statistical approach to line segmentation in handwritten documents. In *Document Recognition and Retrieval XIV*, Xiaofan Lin and Berrin A. Yanikoglu (Eds.). International Society for Optics and Photonics, 65000T.
[17] Nikolaos Arvanitopoulos and Sabine Susstrunk. 2014. Seam carving for text line extraction on color and grayscale historical manuscripts. In *The 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 726–731.
[18] Abedelkadir Asi, Rafi Cohen, Klara Kedem, and Jihad El-Sana. 2015. Simplifying the reading of historical manuscripts. In *The 13th International Conference on Document Analysis and Recognition*. IEEE, 826–830. DOI: https://doi.org/10.1109/ICDAR.2015.7333877
[19] Abedelkadir Asi, Rafi Cohen, Klara Kedem, Jihad El-Sana, and Itshak Dinstein. 2014. A coarse-to-fine approach for layout analysis of ancient manuscripts. In *The 14th International Conference on Frontiers in Handwriting Recognition*. 140–145.
[20] Abedelkadir Asi, Raid Saabni, and Jihad El-Sana. 2011. Text line segmentation for gray scale historical document images. In *The Workshop on Historical Document Imaging and Processing*. ACM Press, New York, 120.
[21] Bruno Tenório Ávila and Rafael Dueire Lins. 2005. A fast orientation and skew detection algorithm for monochromatic document images. In *The ACM Symposium on Document Engineering*. ACM Press, New York, 118.
[22] Micheal Baechler, Marcus Liwicki, and Rolf Ingold. 2013. Text line extraction using DMLP classifiers for historical manuscripts. In *The 12th International Conference on Document Analysis and Recognition*. IEEE, 1029–1033. DOI: https://doi.org/10.1109/ICDAR.2013.206
[23] A. Bagdanov and J. Kanai. 1997. Projection profile based skew estimation algorithm for JBIG compressed images. In *The 4th International Conference on Document Analysis and Recognition*, Vol. 1. IEEE Comput. Soc., 401–405. DOI: https://doi.org/10.1109/ICDAR.1997.619878
[24] Itay Bar-Yosef, Nate Hagbi, Klara Kedem, and Itshak Dinstein. 2009. Line segmentation for degraded handwritten historical documents. In *The 10th International Conference on Document Analysis and Recognition*. IEEE, 1161–1165. http://ieeexplore.ieee.org/document/5277595/.
[25] P. Barlas, S. Adam, C. Chatelain, and T. Paquet. 2014. A typed and handwritten text block segmentation system for heterogeneous and complex documents. In *The 11th IAPR International Workshop on Document Analysis Systems*. IEEE, 46–50.
[26] J. Bernsen. 1986. Dynamic thresholding of gray level images. In *The International Conference on Pattern Recognition*. 1251–1255.
[27] Fadi Biadsy, Jihad El-Sana, and Nizar Habash. 2006. Online Arabic handwriting recognition using hidden Markov models. In *The 10th International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.
[28] Thomas M. Breuel. 2003. High performance document layout analysis. In *Symposium on Document Image Understanding Technology* 3 (2003), 209–218.
[29] D. Bridson and A. Antonacopoulos. 2008. A geometric approach for accurate and efficient performance evaluation of layout analysis methods. In *The 19th International Conference on Pattern Recognition*. IEEE, 1–4.
[30] Syed Saqib Bukhari, Mayce Ibrahim Ali Al Azawi, Faisal Shafait, and Thomas M. Breuel. 2010. Document image segmentation using discriminative learning over connected components. In *The 8th IAPR International Workshop on Document Analysis Systems*. ACM Press, New York, 183–190.
[31] Syed Saqib Bukhari, T. M. Breuel, Abedelkadir Asi, and Jihad El-Sana. 2012. Layout analysis for arabic historical document images using machine learning. In *The International Conference on Frontiers in Handwriting Recognition*. IEEE, 639–644.

[32] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. 2009. Script-independent handwritten textlines segmentation using active contours. In *The 10th International Conference on Document Analysis and Recognition*. IEEE, 446–450. http://ieeexplore.ieee.org/document/5277636/.

[33] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. 2011. Improved document image segmentation algorithm using multiresolution morphology. In *International Society for Optics and Photonics*, Gady Agam and Christian Viard-Gaudin (Eds.). International Society for Optics and Photonics, 78740D.

[34] Marius Bulacu, Rutger Van Koert, Lambert Schomaker, and Tijn van der Zant. 2007. Layout analysis of handwritten historical documents for searching the archive of the cabinet of the Dutch Queen. In *The 9th International Conference on Document Analysis and Recognition*. IEEE, 351–361.

[35] Mark J. Burge and Gladys Monagan. 1995. Using the Voronoi tessellation for grouping words and multipart symbols in documents. In *The SPIE International Symposium on Optics, Imaging and Instrumentation*, Robert A. Melter, Angela Y. Wu, Fred L. Bookstein, and William D. K. Green (Eds.). International Society for Optics and Photonics, 116–124.

[36] C. Clausner A. Antonacopoulos C. Papadopoulos, S. Pletschacher. 2013. The IMPACT dataset of historical document images. In *The 2nd International Workshop on Historical Document Imaging and Processing*. 123–130.

[37] Yang Cao, Shuhua Wang, and Heng Li. 2003. Skew detection and correction in document images based on straight-line fitting. *Pattern Recognition Letters* 24, 12 (2003), 1871–1879.

[38] Samuele Capobianco, Leonardo Scommegna, and Simone Marinai. 2018. Historical handwritten document segmentation by using a weighted loss. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. Springer, 395–406.

[39] R. Cattoni, T. Coianiz, S. Messelodi, and Cm Modena. 1998. Geometric layout analysis techniques for document image understanding: A review. *ITC-First Technical Report* (1998), 1–68.

[40] F. Cesarini, M. Gori, S. Marinai, and G. Soda. 1999. Structured document segmentation and representation by the modified X-Y tree. In *The 5th International Conference on Document Analysis and Recognition*. IEEE, 563–566.

[41] Nabendu Chaki, Soharab Hossain Shaikh, and Khalid Saeed. 2014. A comprehensive survey on image binarization techniques. In *Exploring Image Binarization Techniques*. Springer India, 5–15.

[42] Kai Chen, Cheng-Lin Liu, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2016. Page segmentation for historical document images based on superpixel classification with unsupervised feature learning. In *The 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE, 299–304.

[43] Kai Chen, Mathias Seuret, Jean Hennebert, and Rolf Ingold. 2017. Convolutional neural networks for page segmentation of historical document images. In *The 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 965–970.

[44] Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2015. Page segmentation of historical document images with convolutional autoencoders. In *The 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1011–1015.

[45] Kai Chen, Hao Wei, Jean Hennebert, Rolf Ingold, and Marcus Liwicki. 2014. Page segmentation for historical handwritten document images using color and texture features. In *The 14th International Conference on Frontiers in Handwriting Recognition*. 488–493.

[46] Yiping Chen and Liansheng Wang. 2017. Broken and degraded document images binarization. *Neurocomputing* 237 (2017), 272–280.

[47] Atul K. Chhabra and Ihsin T. Phillips. 1997. The second international graphics recognition contest-raster to vector conversion: A report. In *International Workshop on Graphics Recognition*. Springer, 390–410.

[48] Rafi Cohen, Abedelkadir Asi, Klara Kedem, Jihad El-Sana, and Itshak Dinstein. 2013. Robust text and drawing segmentation algorithm for historical documents. In *The 2nd International Workshop on Historical Document Imaging and Processing*. 110–117.

[49] Rafi Cohen, Itshak Dinstein, Jihad El-Sana, and Klara Kedem. 2014. Using scale-space anisotropic smoothing for text line extraction in historical documents. In *International Conference Image Analysis and Recognition*. Springer International Publishing, 349–358.

[50] Laboratoire National de metrologie et d'Essais (LNE). 2013. MAURDOR campaign. http://www.maurdor-campaign.org/index.php?id=83&L=1.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

[52] Markus Diem, Florian Kleber, and Robert Sablatnig. 2011. Text classification and document layout analysis of paper fragments. In *The International Conference on Document Analysis and Recognition*. IEEE, 854–858. DOI: https://doi.org/10.1109/ICDAR.2011.175

[53] Markus Diem, Florian Kleber, and Robert Sablatnig. 2012. Skew estimation of sparsely inscribed document fragments. In *The 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 292–296.

[54] David Doermann Elena Zotkina, Himanshu Suri. 2013. GEDI: Groundtruthing Environment for Document Images. https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=53.
[55] Boris Epshtein. 2011. Determining document skew using inter-line spaces. In *The International Conference on Document Analysis and Recognition*. IEEE, 27–31. DOI: https://doi.org/10.1109/ICDAR.2011.15
[56] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. 2015. The Delaunay document layout descriptor. In *ACM Symposium on Document Engineering*. ACM Press, New York, 167–175.
[57] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. 2017. A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition* 64 (2017), 1–14.
[58] Jonathan Fabrizio. 2014. A precise skew estimation algorithm for document images using KNN clustering and Fourier transform. In *The International Conference on Image Processing*. IEEE, 2585–2588.
[59] Andreas Fischer, Micheal Baechler, Angelika Garz, Marcus Liwicki, and Rolf Ingold. 2014. A combined system for text line extraction and handwriting recognition in historical documents. In *The 11th IAPR International Workshop on Document Analysis Systems*. 71–75.
[60] Andreas Fischer, Volkmar Frinken, Alicia Fornés, and Horst Bunke. 2011. Transcription alignment of latin manuscripts using hidden Markov models. In *The Workshop on Historical Document Imaging and Processing*. ACM, 29–36.
[61] Gaofeng Meng, Chunhong Pan, Nanning Zheng, and Chen Sun. 2010. Skew estimation of document images using bagging. *IEEE Transactions on Image Processing* 19, 7 (jul 2010), 1837–1846.
[62] Angelika Garz, Markus Diem, and Robert Sablatnig. 2010. Detecting text areas and decorative elements in ancient manuscripts. In *The 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 176–181. DOI: https://doi.org/10.1109/ICFHR.2010.35
[63] Angelika Garz, Andreas Fischer, Robert Sablatnig, and Horst Bunke. 2012. Binarization-free text line segmentation for historical documents based on interest point clustering. In *The 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 95–99.
[64] Angelika Garz and Robert Sablatnig. 2010. Multi-scale texture-based text recognition in ancient manuscripts. In *The 16th International Conference on Virtual Systems and Multimedia*. IEEE, 336–339. DOI: https://doi.org/10.1109/VSMM.2010.5665938
[65] Angelika Garz, Robert Sablatnig, and Markus Diem. 2011. Layout analysis for historical manuscripts using SIFT features. In *The International Conference on Document Analysis and Recognition*. 508–512.
[66] B. Gatos, N. Papamarkos, and C. Chamzas. 1997. Skew detection and text line position determination in digitized documents. *Pattern Recognition* 30, 9 (1997), 1505–1519.
[67] B. Gatos, N. Stamatopoulos, and G. Louloudis. 2011. ICDAR2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition (IJDAR)* 14, 1 (2011), 25–33.
[68] Basilios Gatos, Pratikakis Ioannis, and Stavros J. Perantonis. 2004. An adaptive binarization technique for low quality historical documents. In *Document Analysis Systems VI*. Springer, Springer Berlin, 102–113.
[69] Basilis Gatos, Nikolaos Stamatopoulos, and Georgios Louloudis. 2010. ICFHR2010 handwriting segmentation contest. In *The 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 737–742.
[70] Tobias Grüning, Gundram Leifert, Tobias Strauß, and Roger Labahn. 2018. A two-stage method for text line detection in historical documents. *arXiv preprint arXiv:1802.03345* (2018).
[71] Karim Hadjar and Rolf Ingold. 2004. Physical layout analysis of complex structured arabic documents using artificial neural nets. In *Lecture Notes in Computer Science*. Springer Berlin, 170–178.
[72] Sheng He and Lambert Schomaker. 2019. DeepOtsu: Document enhancement and binarization using iterative deep learning. *Pattern Recognition* 91 (2019), 379–390.
[73] S. C. Hinds, J. L. Fisher, and D. P. D'Amato. 1990. A document skew detection method using run-length encoding and the hough transform. In *The 10th International Conference on Pattern Recognition*, Vol. I. IEEE Comput. Soc. Press, 464–468.
[74] Jaekyu Ha, R. M. Haralick, and I. T. Phillips. 1995. Document page decomposition by the bounding-box project. In *The 3rd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press, 1119–1122.
[75] Anil K. Jain and Yu Zhong. 1996. Page segmentation using texture analysis. *Pattern Recognition* 29, 5 (May 1996), 743–770.
[76] N. Journet, V. Eglin, J. Y. Ramel, and R. Mullot. 2005. Text/graphic labelling of ancient printed documents. In *The 8th International Conference on Document Analysis and Recognition*. IEEE, 1010–1014 Vol. 2. DOI: https://doi.org/10.1109/ICDAR.2005.235
[77] Nicholas Journet, Jean-Yves Ramel, Rémy Mullot, and Véronique Eglin. 2008. Document image characterization using a multiresolution analysis of the texture: Application to old documents. *International Journal of Document Analysis and Recognition (IJDAR)* 11, 1 (Jun 2008), 9–18.

[78] Hao Wei Marcus Liwicki Rolf Ingold Kai Chen, Mathias Seuret. 2015. Document, image, and video analysis DLA tool. http://diuf.unifr.ch/main/hisdoc/divadia.

[79] Rangachar Kasturi, Lawrence O'Gorman, and Venu Govindaraju. 2002. Document image analysis: A primer. *Sadhana* 27, 1 (2002), 3–22.

[80] N. Khorissi, A. Namane, A. Mellit, F. Abdati, Z. A. Bensalama, and A. Guessoum. 2007. Application of the wavelet and the Hough transform for detecting the skew angle in arabic printed documents. In *The 9th International Symposium on Signal Processing and Its Applications*. IEEE, 1–4. http://ieeexplore.ieee.org/document/4555586/.

[81] Koichi Kise, Akinori Sato, and Motoi Iwata. 1998. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding* 70, 3 (1998), 370–382.

[82] Koichi Kise. 2014. Page segmentation techniques in document analysis. In *Handbook of Document Image Processing and Recognition*. Springer London, London, 135–175.

[83] K. Kise, A. Sato, and K. Matsumoto. 1997. Document image segmentation as selection of Voronoi edges. In *The Workshop on Document Image Analysis*. IEEE Comput. Soc, 32–39. DOI: https://doi.org/10.1109/DIA.1997.627089

[84] Florian Kleber, Robert Sablatnig, Melanie Gau, and Heinz Miklas. 2008. Ancient document analysis based on text line extraction. In *The 19th International Conference on Pattern Recognition*. IEEE, 1–4. DOI: https://doi.org/10.1109/ICPR.2008.4761530

[85] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. 1993. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 7 (1993), 737–747.

[86] Victor Lavrenko, Toni M. Rath, and Raghavan Manmatha. 2004. Holistic word recognition for handwritten historical documents. In *The 1st International Workshop on Document Image Analysis for Libraries*. IEEE, 278–287.

[87] Daniel S. Le, George R. Thoma, and Harry Wechsler. 1994. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition* 27, 10 (1994), 1325–1344.

[88] Shutao Li, Qinghua Shen, and Jun Sun. 2007. Skew detection using wavelet decomposition and projection profile analysis. *Pattern Recognition Letters* 28, 5 (2007), 555–562.

[89] L. Likforman-Sulem, A. Hanimyan, and C. Faure. 1995. A Hough based algorithm for extracting text lines in handwritten documents. In *The 3rd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press, 774–777.

[90] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. 2007. Text line segmentation of historical documents: A survey. *International Journal of Document Analysis and Recognition (IJDAR)* 9, 2–4 (Sept. 2007), 123–138. http://link.springer.com/10.1007/s10032-006-0023-z

[91] N. Liolios, N. Fakotakis, and G. Kokkinakis. 2001. Improved document skew detection based on text line connected-component clustering. In *The International Conference on Image Processing (Cat. No.01CH37205)*, Vol. 1. IEEE, 1098–1101.

[92] G. Louloudis, B. Gatos, and C. Halatsis. 2007. Text line detection in unconstrained handwritten documents using a block-based Hough transform approach. In *The 9th International Conference on Document Analysis and Recognition*. IEEE, 599–603.

[93] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. 2009. Text line and word segmentation of handwritten documents. *Pattern Recognition* 42, 12 (2009), 3169–3183.

[94] Scott Lowther, Vinod Chandran, and Subramanian Sridharan. 2002. An accurate method for skew determination in document images. In *Digital Image Computing Techniques and Applications*, Vol. 1. 25–29.

[95] Yue Lu and Chew Lim Tan. 2003. A nearest-neighbor chain based approach to skew estimation in document images. *Pattern Recognition Letters* 24, 14 (2003), 2315–2323.

[96] Yue Lu, Zhe Wang, and Chew Lim Tan. 2004. Word grouping in document images based on Voronoi tessellation. In *International Workshop on Document Analysis Systems*. Springer Berlin, 147–157.

[97] Simon M. Lucas. 2005. ICDAR 2005 text locating competition results. In *The 8th International Conference on Document Analysis and Recognition*. IEEE, 80–84.

[98] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. 2003. Document structure analysis algorithms: A literature survey. *SPIE 5010, Document Recognition and Retrieval X* 5010, 1 (2003), 197.

[99] Simone Marinai, Marco Gori, and Giovanni Soda. 2005. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1 (2005), 23–35.

[100] Gale L. Martin. 1993. Centered-object integrated segmentation and recognition of overlapping handprinted characters. *Neural Computation* 5, 3 (1993), 419–429.

[101] Maroua Mehri, Petra Gomez-Krämer, Pierre Héroux, Alain Boucher, and Rémy Mullot. 2013. Texture feature evaluation for segmentation of historical document images. In *The 2nd International Workshop on Historical Document Imaging and Processing*. ACM Press, New York, 102.

[102] Maroua Mehri, Pierre Héroux, Petra Gomez-Krämer, and Rémy Mullot. 2017. Texture feature benchmarking and evaluation for historical document image analysis. *International Journal on Document Analysis and Recognition (IJDAR)* 20, 1 (2017), 1–35.

[103] Maroua Mehri, Nibal Nayef, Pierre Héroux, Petra Gomez-Krämer, and Rémy Mullot. 2015. Learning texture features for enhancement and segmentation of historical document images. In *The 3rd International Workshop on Historical Document Imaging and Processing*. ACM Press, New York, 47–54.

[104] G. Nagy. 2000. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1 (2000), 38–62.

[105] George Nagy and Sharad Seth. 1984. Hierarchical representation of optically scanned documents. In *The International Conference on Pattern Recognition*. IEEE, 347–349.

[106] Y. Nakano, Y. Shima, H. Fujisawa, J. Higashino, and M. Fujinawa. 1990. An algorithm for the skew normalization of document image. In *The 10th International Conference on Pattern Recognition*, Vol. 2. IEEE Comput. Soc. Press, 8–13.

[107] N. Nandini, K. Srikanta Murthy, and G. Hemantha Kumar. 2008. Estimation of skew angle in binary document images using hough transform. *World Academy of Science, Engineering and Technology* 18 (2008), 44–49.

[108] Wayne; Niblack. 1986. *An Introduction to Digital Image Processing*. Prentice-Hall, Englewood Cliffs NJ. 115–116 pages.

[109] Nikos Nikolaou, Michael Makridis, Basilis Gatos, Nikolaos Stamatopoulos, and Nikos Papamarkos. 2010. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing* 28, 4 (Apr. 2010), 590–604.

[110] Konstantinos Ntirogiannis, Basilis Gatos, and Ioannis Pratikakis. 2014. ICFHR2014 competition on handwritten document image binarization (H-DIBCO 2014). In *The 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 809–813.

[111] L. O'Gorman. 1993. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 11 (1993), 1162–1173.

[112] Oleg Okun, Matti Pietikäinen, O. Okun, and M. Pietikäinen. 1999. A survey of texture-based methods for document layout analysis. In *Workshop on Texture Analysis in Machine Vision*. 137–148.

[113] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhSegment: A generic deep-learning approach for document segmentation. In *The 16th International Conference on Frontiers in Handwriting Recognition*. IEEE, 7–12.

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

[115] U. Pal and B. B. Chaudhuri. 1996. An improved document skew angle estimation technique. *Pattern Recognition Letters* 17, 8 (1996), 899–904.

[116] G. S. Peake and T. N. Tan. 1997. A general algorithm for document skew angle estimation. In *The International Conference on Image Processing*. IEEE Comput. Soc., 230–233.

[117] Ihsin T. Phillips and Atul K. Chhabra. 1999. Empirical performance evaluation of graphics recognition systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 9 (1999), 849–870.

[118] Ihsin T. Phillips, Jisheng Liang, Atul K. Chhabra, and Robert Haralick. 1997. A performance evaluation protocol for graphics recognition systems. In *International Workshop on Graphics Recognition*. Springer, 372–389.

[119] Stefan Pletschacher and Apostolos Antonacopoulos. 2010. The PAGE (page analysis and ground-truth elements) format framework. In *The 20th International Conference on Pattern Recognition*. IEEE, 257–260.

[120] Wolfgang Postl. 1986. Detection of linear oblique structures and skew scan in digitized documents. In *The 8th International Conference on Pattern Recognition*. 687–689.

[121] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. 2016. ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016). In *The 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 619–623.

[122] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. 2017. ICDAR2017 competition on document image binarization (DIBCO 2017). In *The 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 1395–1403.

[123] Lorenzo Quirós. 2018. Multi-task handwritten document layout analysis. *arXiv preprint arXiv:1806.08852* (2018).

[124] Lorenzo Quirós, Lluís Serrano, Vicente Bosch, Alejandro H. Toselli, Rosa Congost, Enric Saguer, and Enrique Vidal. 2018. HTR Dataset ICFHR 2018. https://zenodo.org/record/1322666#.XHOanOgzaUk.

[125] Irina Rabaev, Ofer Biller, Jihad El-Sana, Klara Kedem, and Itshak Dinstein. 2013. Text line detection in corrupted and damaged historical manuscripts. In *The 12th International Conference on Document Analysis and Recognition*. IEEE, 812–816.

[126] J. Y. Ramel, S. Leriche, M. L. Demonet, and S. Busson. 2007. User-driven page layout analysis of historical printed books. *International Journal of Document Analysis and Recognition (IJDAR)* 9, 2–4 (Apr. 2007), 243–261.

[127] Marte A. Ramírez-Ortegón, Lilia L. Ramírez-Ramírez, Ines Ben Messaoud, Volker Märgner, Erik Cuevas, and Raúl Rojas. 2014. A model for the gray-intensity distribution of historical handwritten documents and its application for binarization. *International Journal on Document Analysis and Recognition* 17, 2 (2014), 139–160.

[128] Tony M. Rath and Rudrapatna Manmatha. 2007. Word spotting for historical documents. *International Journal on Document Analysis and Recognition* 9, 2 (2007), 139–152.

[129] Ahsen Raza, Imran Siddiqi, Ali Abidi, and Fahim Arif. 2012. An unconstrained benchmark urdu handwritten sentence database with automatic line segmentation. In *International Conference on Frontiers in Handwriting Recognition*. IEEE, 491–496.

[130] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 234–241.

[131] Raid Saabni, Abedelkadir Asi, and Jihad El-Sana. 2014. Text line extraction for historical document images. *Pattern Recognition Letters* 35, 1 (2014), 23–33.

[132] Raid Saabni and Jihad El-Sana. 2011. Language-independent text lines extraction using seam carving. In *The International Conference on Document Analysis and Recognition*. IEEE, 563–568.

[133] Rana S. M. Saad, Randa I. Elanwar, N. S. Abdel Kader, Samia Mashali, and Margrit Betke. 2016. BCE-Arabic-v1 dataset: Towards interpreting arabic document images for people with visual impairments. In *The 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA*. ACM Press, New York, New York, USA, 1–8.

[134] T. Saitoh, M. Tachikawa, and T. Yamaai. 1993. Document image segmentation and text area ordering. In *The 2nd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press, 323–329.

[135] P. Saragiotis and N. Papamarkos. 2008. Local skew correction in documents. *International Journal of Pattern Recognition and Artificial Intelligence* 22, 4 (2008), 691–710.

[136] M. Sarfraz, S. A. Mahmoud, and Z. Rasheed. 2007. On skew estimation and correction of text. In *Computer Graphics, Imaging and Visualisation*. IEEE, 308–313.

[137] Eric Saund, Jing Lin, and Prateek Sarkar. 2009. PixLabeler: User interface for pixel-level labeling of elements in document images. In *The 10th International Conference on Document Analysis and Recognition*. IEEE, 646–650.

[138] J. Sauvola and M. Pietikainen. 1995. Skew angle detection using texture direction analysis. In *The 9th Scandinvian Conference on Image Analysis*. 1099–1106.

[139] J. Sauvola and M. Pietikäinen. 2000. Adaptive document image binarization. *Pattern Recognition* 33, 2 (2000), 225–236.

[140] Seong-Whan Seong-Whan Lee and Dae-Seok Dae-Seok Ryu. 2001. Parameter-free geometric document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1240–1256.

[141] Mathias Seuret, Michele Alberti, Marcus Liwicki, and Rolf Ingold. 2017. PCA-initialized deep neural networks applied to document image analysis. In *The 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 877–882.

[142] F. Shafait and T. M. Breuel. 2011. The effect of border noise on the performance of projection-based page segmentation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 4 (2011), 846–851.

[143] F. Shafait, D. Keysers, and T. M. Breuel. 2006. Pixel-accurate representation and evaluation of page segmentation in document images. In *The 18th International Conference on Pattern Recognition*. IEEE, 872–875. DOI: https://doi.org/10.1109/ICPR.2006.934

[144] Faisal Shafait, Joost van Beusekom, Daniel Keysers, and Thomas M. Breuel. 2008. Background variability modeling for statistical layout analysis. In *The 19th International Conference on Pattern Recognition*. IEEE, 1–4.

[145] Mahnaz Shafii and Maher Sid-Ahmed. 2015. Skew detection and correction based on an axes-parallel bounding box. *International Journal on Document Analysis and Recognition (IJDAR)* 18, 1 (2015), 59–71.

[146] Asif Shahab. 2013. UW3 and UNLV Datasets. Http://www.iapr-tc11.org/mediawiki/index.php/Table_Ground_Truth_for_the_UW3_and_UNLV_datasets.

[147] Zhixin Shi and Venu Govindaraju. 2004. Line separation for complex document images using fuzzy runlength. In *The 1st International Workshop on Document Image Analysis for Libraries*. 306–312.

[148] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. 2009. A steerable directional local profile technique for extraction of handwritten Arabic text lines. In *The 10th International Conference on Document Analysis and Recognition*. IEEE, 176–180.

[149] Frank Y. Shih and Shy-Shyan Chen. 1996. Adaptive document block segmentation and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26, 5 (1996), 797–802.

[150] P. Shivakumara, G. Hemantha Kumar, D. S. Guru, and P. Nagabhushan. 2005. A novel technique for estimation of skew in binary text document images based on linear regression analysis. *Sadhana* 30, 1 (2005), 69–85.

[151] Fotini Simistira, Manuel Bouillon, Mathias Seuret, Marcel Wursch, Michele Alberti, Rolf Ingold, and Marcus Liwicki. 2017. ICDAR2017 competition on layout analysis for challenging medieval manuscripts. In *The 14th IAPR International Conference on Document Analysis and Recognition*. IEEE, 1361–1370.

[152] A. Simon, J.-C. Pret, and A. P. Johnson. 1997. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 3 (1997), 273–277.

[153] Brij Mohan Singh, Rahul Sharma, Debashis Ghosh, and Ankush Mittal. 2014. Adaptive binarization of severely degraded and non-uniformly illuminated documents. *International Journal on Document Analysis and Recognition (IJDAR)* 17, 4 (2014), 393–412.

[154] Chandan Singh, Nitin Bhatia, and Amandeep Kaur. 2008. Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition* 41, 12 (2008), 3528–3546.
[155] Bolan Su, Shijian Lu, and Chew Lim Tan. 2010. Binarization of historical document images using the local maximum and minimum. In *The 8th IAPR International Workshop on Document Analysis Systems*. ACM Press, New York, 159–166.
[156] Wassim Swaileh, Kamel Ait Mohand, and Thierry Paquet. 2015. Multi-script iterative steerable directional filtering for handwritten text line extraction. In *The 13th International Conference on Document Analysis and Recognition*. IEEE, 1241–1245.
[157] D. Sylwester and S. Seth. 1995. A trainable, single-pass algorithm for column segmentation. In *The 3rd International Conference on Document Analysis and Recognition*, Vol. 2. IEEE Comput. Soc. Press, 615–618.
[158] Breuel Thomas and Faisal Shafait. 2010. AutoMLP: Simple, effective, fully automated learning rate and size adjustment. In *The Learning Workshop, Utah*.
[159] Tuan Anh Tran, In-Seop Na, and Soo-Hyung Kim. 2015. Hybrid page segmentation using multilevel homogeneity structure. In *The 9th International Conference on Ubiquitous Information Management and Communication*. ACM Press, New York, 1–6.
[160] Tuan Anh Tran, In Seop Na, and Soo Hyung Kim. 2016. Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology. *International Journal on Document Analysis and Recognition (IJDAR)* 19, 3 (Sep. 2016), 191–209.
[161] Nikos Vasilopoulos and Ergina Kavallieratou. 2017. Complex layout analysis based on contour classification and morphological operations. *Engineering Applications of Artificial Intelligence* 65 (2017), 220–229.
[162] Friedrich M. Wahl, Kwan Y. Wong, and Richard G. Casey. 1982. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing* 20, 4 (Dec. 1982), 375–390.
[163] Hao Wei, Micheal Baechler, Fouad Slimane, and Rolf Ingold. 2013. Evaluation of SVM, MLP and GMM classifiers for layout analysis of historical documents. In *The 12th International Conference on Document Analysis and Recognition*. IEEE, 1220–1224.
[164] Hao Wei, Kai Chen, Rolf Ingold, and Marcus Liwicki. 2014. Hybrid feature selection for historical document layout analysis. In *The 14th International Conference on Frontiers in Handwriting Recognition*. 87–92.
[165] Hao Wei, Kai Chen, Anguelos Nicolaou, Marcus Liwicki, and Rolf Ingold. 2014. Investigation of feature selection for historical document layout analysis. In *The 4th International Conference on Image Processing Theory, Tools and Applications*. 1–6.
[166] Florian Westphal, Niklas Lavesson, and Håkan Grahn. 2018. Document image binarization using recurrent neural networks. In *The 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 263–268.
[167] Christoph Wick and Frank Puppe. 2018. Fully convolutional neural networks for page segmentation of historical document images. In *The 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 287–292.
[168] Chung-Chih Wu, Chien-Hsing Chou, and Fu Chang. 2008. A machine-learning approach for analyzing document layout structures with two reading orders. *Pattern Recognition* 41, 10 (2008), 3200–3213.
[169] Yi Xiao and Hong Yan. 2003. Text region extraction in a document image based on the Delaunay tessellation. *Pattern Recognition* 36, 3 (2003), 799–809.
[170] Yi Xiao and Hong Yan. 2004. Location of title and author regions in document images based on the Delaunay triangulation. *Image and Vision Computing* 22, 4 (2004), 319–329.
[171] H. Yan. 1993. Skew correction of document images using interline cross-correlation. *Graphical Models and Image Processing* 55, 6 (1993), 538–543.
[172] Younki Min, Sung-Bae Cho, and Yillbyung Lee. 1996. A data reduction method for efficient document skew estimation based on Hough transformation. In *The 13th International Conference on Pattern Recognition*, Vol. 3. IEEE, 732–736.
[173] Bin Yu and Anil K. Jain. 1996. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition* 29, 10 (Oct. 1996), 1599–1629.
[174] Yue Lu and C. L. Tan. 2005. Constructing area Voronoi diagram in document images. In *The 8th International Conference on Document Analysis and Recognition*, Vol. 1. IEEE, 342–346.
[175] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane. 2001. Arabic hand-written text-line extraction. In *The 6th International Conference on Document Analysis and Recognition*. IEEE Comput. Soc., 281–285.
[176] Yefeng Zheng and David Doermann. 2010. LAMP Dataset of Layer Separation. https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=61.

Received July 2018; revised July 2019; accepted July 2019