Received: 8 June 2023 | Revised: 23 October 2023 | Accepted: 28 November 2023

DOI: 10.1002/eng2.12832

RESEARCH ARTICLE

Engineering Reports Open Access WILEY

Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection

Amirreza Fateh[1] | **Mansoor Fateh**[2] | **Vahid Abolghasemi**[3]

[1]School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran

[2]Faculty of Computer Engineering, Shahrood University of Technology, Shahrud, Iran

[3]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

Correspondence
Mansoor Fateh, Faculty of Computer Engineering, Shahrood University of Technology, Shahrud, Iran.
Email: mansoor_fateh@shahroodut.ac.ir

Abstract
In recent years, automatic document and text analysis has gained significant importance, driven by advancements in optical character recognition (OCR) technology and the need for efficient processing of large volumes of printed or handwritten documents. This article specifically focuses on document layout analysis (DLA) and text line detection (TLD), both of which are crucial components of OCR systems. Our objective is to develop an effective method for extracting both textual and non-textual regions, addressing challenges unique to the Persian (and Persian-like) language(s). In the DLA stage, we employ deep learning models and a voting system to accurately determine the regions of interest. Additionally, we introduce methods such as optimum font size concepts, angle correction, and a line curvature elimination algorithm in the TLD process to enhance OCR accuracy. Comparative evaluations against state-of-the-art methods demonstrate the superiority of our approach, showcasing a 2.8% improvement in the accuracy of Tesseract-OCR 5.1.0 (a well-established commercial OCR system) on the official Iranian newspapers dataset. These findings underscore the importance of addressing DLA and TLD challenges to advance OCR technology for Persian language documents and provide a solid foundation for future research in this domain.

KEYWORDS
connected component, document layout analysis, font size, line detection, Persian printed

1 | INTRODUCTION

Automatic document and text analysis has become an essential component of many artificially intelligent systems in recent years. With the increased availability of processing power, OCR technology has made it possible to quickly and accurately analyse massive amounts of printed or handwritten documents.[1,2] Furthermore, the application of explainable artificial intelligence (XAI) techniques in OCR systems has gained prominence in recent research. XAI methods enhance the interpretability of OCR outcomes, shedding light on how the system recognizes and interprets textual content. These techniques have found valuable applications in various domains, including handwritten recognition and image spam detection, providing insights into the decision-making processes of OCR algorithms.[3,4]

OCR systems typically comprise three main steps: document layout analysis (DLA), text line detection (TLD), and optical character recognition (OCR). In the first step, DLA, the system separates and extracts textual and non-textual areas from the image. This step is essential because it allows the OCR system to focus solely on the text areas within the image, thereby enhancing the accuracy of the recognition process.[5,6] In the second step, TLD, the OCR system extracts individual lines of text from the image. This stage is critical as it ensures that each line is accurately recognized and processed.[7,8] Finally, in the third step, the OCR system analyzes each line and converts it into editable characters. Achieving precise and reliable OCR results necessitates meticulous attention to each of these stages, as errors at any point in the process can significantly diminish the accuracy of the final output.

OCR systems heavily depend on the quality of the DLA and TLD steps, as the accuracy of character detection is directly influenced by the success rate of these stages. Nevertheless, DLA and TLD modules are computationally intensive and require a substantial number of pixels to be processed.[9] To develop an efficient OCR software, it is essential to design DLA and TLD modules optimized for both performance and resource utilization. However, images obtained from various imaging systems may exhibit distortions, such as blur, low resolution, and uneven illumination. These distortions can significantly undermine OCR accuracy if not adequately addressed during the preprocessing stage.[10]

One major challenge in DLA systems is the absence of a standardized format for all pages. While some pages may lack unique complexities, such as irregular shapes, tables, or graphs, others may contain these elements.[11] Complex backgrounds, noise, low image quality, and image rotation can further complicate the detection and extraction of text-containing regions. In TLD, dealing with curved lines poses a significant challenge that can significantly impact OCR accuracy. When a word's baseline does not align with the line's baseline, OCR performance can deteriorate.[7] Therefore, an efficient OCR system should be capable of accommodating all page formats and strive to minimize line curvature as much as possible to mitigate OCR difficulties.

Finally, character and TLD approaches cannot be universally applied to scripts with different languages. This challenge is particularly pronounced in languages such as Persian and Arabic, where text characters can take the form of connectors or non-connectors. Connector letters are affixed to both pre- and post-letters to form words, and diacritic marks are commonly used in Arabic text—both of which can introduce complexity to TLD techniques.[12] Consequently, detecting and extracting text in these languages presents a formidable problem. While several TLD techniques have been developed for Latin script languages, further research is necessary to devise efficient methods tailored to non-Latin languages, such as Persian.

This study addresses prevalent challenges in OCR and introduces an innovative method designed to enhance system accuracy and efficiency. Our approach comprises two fundamental components: DLA and TLD. The DLA module focuses on accurately identifying the document's layout structure, encompassing the arrangement of text blocks, images, and visual elements. Its key role lies in segregating the main text, providing a structured representation of the document's content. Conversely, the TLD module precisely detects and segments individual text lines, facilitating precise character recognition in subsequent OCR pipeline stages. Both DLA and TLD steps are carefully crafted to optimize performance and resource utilization for efficient processing of documents with varying complexities.

In this research, we present a new OCR method that propels OCR systems to higher levels of performance through groundbreaking innovations. First, we introduce the official Iranian newspapers (OIN) dataset, a curated collection of 1920 newspaper images, valuable for training and evaluating OCR models, especially in challenging scenarios.[13] In the DLA step, our method employs a smart voting system—a unique approach that combines the strengths of four deep learning-based methods, enhancing accuracy in layout structure identification. This voting system uses a 5×5 window mechanism for both textual and non-textual regions, representing a pivotal advancement in DLA and distinguishing our method from conventional approaches. In the TLD step, we introduce an innovative approach that begins with recognizing optimum font sizes using a distance transform, setting the stage for improved accuracy. Additionally, we employ an angle correction technique with low complexity, significantly enhancing efficiency. Notably, our TLD method innovatively addresses the challenge of eliminating line curvature, a seldom-tackled issue in existing techniques. These contributions collectively exemplify the advancements our method brings to OCR, offering superior accuracy, efficiency, and robustness in extracting textual information from a diverse range of documents. Through comprehensive comparisons with state-of-the-art deep learning methods in both DLA and TLD steps, we empirically demonstrate the substantial performance gains our approach achieves.

2 | RELATED WORKS

OCR is a pivotal technology that enables the extraction of textual information from images or scanned documents, facilitating its conversion into machine-readable formats.[7] The OCR pipeline typically encompasses three key steps: DLA, TLD, and OCR. For the OCR step, we utilized Tesseract-OCR 5.1.0, a widely adopted tool for Farsi OCR. Notably, previous research has extensively covered the usage of Tesseract-OCR versions for OCR in Farsi, thus rendering it beyond the scope of our current work. This section focuses on exploring the state-of-the-art advancements in DLA and TLD, which serve as integral components within OCR systems.

2.1 | Document layout analysis methods

With the advancement of science and technology, the efficiency of artificial intelligence (AI)-based systems has significantly increased. Due to the growing usage of these systems, the demands placed on them have also escalated. Therefore, artificial intelligence systems must continuously improve to meet user needs. DLA systems, and consequently OCR systems, are no exception to this rule. In general, algorithms related to DLA are relevant to this research. There are various techniques and approaches utilized in DLA, including rule-based methods, statistical models, and machine-learning algorithms. In this subsection, we will examine some of the research conducted in the field of DLA.

In recent years, significant research has focused on improving DLA systems through integrating advanced techniques. Some of the critical areas of investigation include the utilization of deep learning models, such as convolutional neural networks (CNNs), to enhance the accuracy and robustness of layout analysis.[13–16] These models have demonstrated promising results in automatically detecting and segmenting different document components.

One of the novel deep learning methods is the YOLO (You Only Look Once) method.[13] First introduced in 2016, this method is considered one of the real-time object detection methods and has shown better performance compared to other object detection methods. YOLOv3[17] is a method based on YOLO, consisting of 53 trained layers on ImageNet and an additional 53 layers for object detection. Due to the 106 layers in YOLOv3, this architecture is slightly slower than other YOLO structures like YOLOv2, but it provides higher accuracy and better performance.

Another suitable method for detecting text and non-text regions is faster R-CNN (region-based convolutional neural networks).[14] The main difference between this method and previous methods like R-CNN and fast R-CNN lies in how regions are extracted. While the previous methods use selective search to extract desired regions, faster R-CNN introduces region proposal networks (RPN).[14]

Another deep learning-based method is single shot multibox detector (SSD).[15] This method utilizes a convolutional network to compute the feature map of the input image. Then, by applying small 3×3 convolutional networks on the feature map, text and non-text regions are detected, and bounding boxes are drawn around them.

Layout Parser is another method used for detecting text-containing regions.[16] Layout Parser is a deep learning-based tool for DLA tasks. This tool, accessible via GitHub, is a relatively powerful tool capable of extracting text and non-text regions from complex images.[16]

By examining the advancements in DLA research, we gain valuable insights into the various techniques and methodologies employed to improve the performance of OCR systems. The continuous development and refinement of DLA algorithms contribute to OCR systems' overall effectiveness and usability in extracting textual information from images or scanned documents.

In the realm of DLA, numerous methodologies have been proposed to tackle the challenges of text and non-text region identification. These methods have been extensively evaluated on a range of standard datasets to assess their effectiveness. Notably, several widely recognized and standard datasets have played a pivotal role in evaluating DLA methods, ensuring the robustness and applicability of the proposed techniques.

Prominent among these datasets is the PRImA dataset, which represents magazine layouts,[18] and PubLayNet, which focuses on academic paper layouts.[19] Additionally, the Table Bank dataset has been instrumental in the assessment of DLA methods, particularly for the extraction of tables in academic papers.[20] For documents with complex layout structures, such as newspaper figures, the Newspaper Navigator dataset has been a valuable resource.[21] Furthermore, the HJDataset, which encompasses historical Japanese document layouts, has served as a critical benchmark for methods aiming to address unique challenges posed by historical texts.[22]

It is noteworthy that the DLA methods discussed in this section have been extensively trained and evaluated on these standard datasets. This rigorous evaluation process ensures that the proposed DLA techniques are not only effective but also adaptable to diverse document types and layouts.

2.2 Text line detection methods

TLD has been an essential part of most document analysis systems, such as OCR, text restoration, and binarization. According to the literature, TLD techniques are divided into seven different categories: (1) CC-based,[9,23] (2) project profile-based,[24–26] (3) smearing-based,[12,25] (4) bounding-based,[25] (5) Hough transform,[27] (6) combined,[25,28] and (7) deep learning-based methods.[29–33] In what follows, these relevant techniques are reviewed.

In Reference 34, a camera-captured document image analysis was proposed with the focus on TLD stage. The method is based on CCs and is combined with an extended version of scale selection method called state estimation. The method first extracts CCs using the maximally stable extremal region algorithm. Then, scale and orientation of CCs are calculated from the associated projection profiles. Finally, a binary classifier is trained to recognize text-line and non-text-line components.

In Reference 23, edge information, extracted from CCs, are used to perform TLD on typewritten script in Urdu language. The authors apply a sequence of preprocessing steps on the input text image. Then, the edge information is extracted from connected components to segment the lines, words and characters. The method was tested with two local datasets collected by the authors. The average reported accuracy is 86.05%.

A different approach for TLD based on CCs was taken in Reference 9. The authors first extract the CCs from the document image. Then, the direction of text-line in multiple local regions is estimated. This is proposed in form of an optimization problem which is solved mathematically. Then, a graph is constituted with nodes as symbols (CCs) and edges as the estimated text-line directions. Finally, the graph is partitioned according to the nodes' connectives. The experimental results show robustness of this method against non-uniform skew within text lines and font sizes variability.

Projection profiling is considered as another common approach for TLD. It is mainly performed in two ways: horizontal or vertical.[24–26] In horizontal projection profiling the interline gap between lines are to be found. This gap is sought as a separator between two consequent lines. Vertical projection profiling operates in a similar manner but vertically. It is more suitable for character detection. Both these techniques are suitable for typed text, where there is neither overlapping nor touching between lines and/or characters.[25] There are, however, few works that address projection profiling for texts with overlapping and touching text, for example, a strip-based method proposed in Reference 26.

There are some studies in the literature that perform TLD using the so-called smearing techniques. These techniques attempt to segment the text-lines through generating similar regions along with the text locations.[12,35] Smearing techniques have shown success with both type-written and hand-written texts. Nevertheless, they often fail when the text includes overlapping or touching between the lines.[12,25]

Deep learning techniques have recently shown promising performance in TLD. Deep neural network models require large-scale training datasets to achieve a high segmentation accuracy. This is a significant limitation in some languages like Persian, where such a set is unavailable. In a study by Lyu Bing et al.,[29] the ARU-Net[33] was used for TLD. ARU-Net is a deep learning model for TLD in historical documents. It encompasses A-Net and RU-Net as core networks.

Another promising work was conducted in Reference 36 based on a neural network called OCRopus. This approach was later extended and improved by several researchers. Based on the most recent version in 2017, line detection and OCR are performed using LSTM and statistical linguistic models, respectively.[32]

A turn-key OCR system, called Kraken, optimized for historical and non-Latin scripts has been recently proposed as an open-source platform.[31] Kraken performs TLD and character recognition using a deep neural network. Kraken's main features are word bounding boxes and character cuts, variable recognition network architectures, and multi-script recognition support.

TLD in complex scripts like Persian and Arabic presents unique challenges, and the availability of suitable standard datasets is limited. Unlike English or other widely studied languages, Persian and Arabic require specialized datasets and approaches to tackle text line extraction effectively.

While some TLD methods have utilized publicly available datasets, such as Arabic datasets for Arabic script-based work, the scarcity of Persian or Arabic-specific datasets remains a hurdle. Notably, Urdu, a language with some similarities to Persian, has been leveraged for TLD research. For instance,[24] employed the Urdu Printed Text Images (UPTI) dataset, comprising an impressive 189,000 ligatures, demonstrating the suitability of Urdu datasets for script-agnostic TLD approaches.[37]

However, some TLD methods have had to create their datasets due to the lack of publicly available resources. For instance, Garg et al.,[26] focusing on the Punjabi script, exemplifies this approach by generating their dataset. Additionally, certain methods like Soujanya et al.,[25] which explored English and Telugu languages, resorted to self-created datasets to train and validate their models. Furthermore, the majority of deep learning-based TLD methods predominantly rely on English datasets, underscoring the need for dedicated datasets for Persian and Arabic languages. Recognizing this need, we have introduced the OIN dataset in this work to facilitate research in Persian language document analysis, which further contributes to the advancement of OCR technology for non-Latin languages.

3 | PROPOSED METHOD

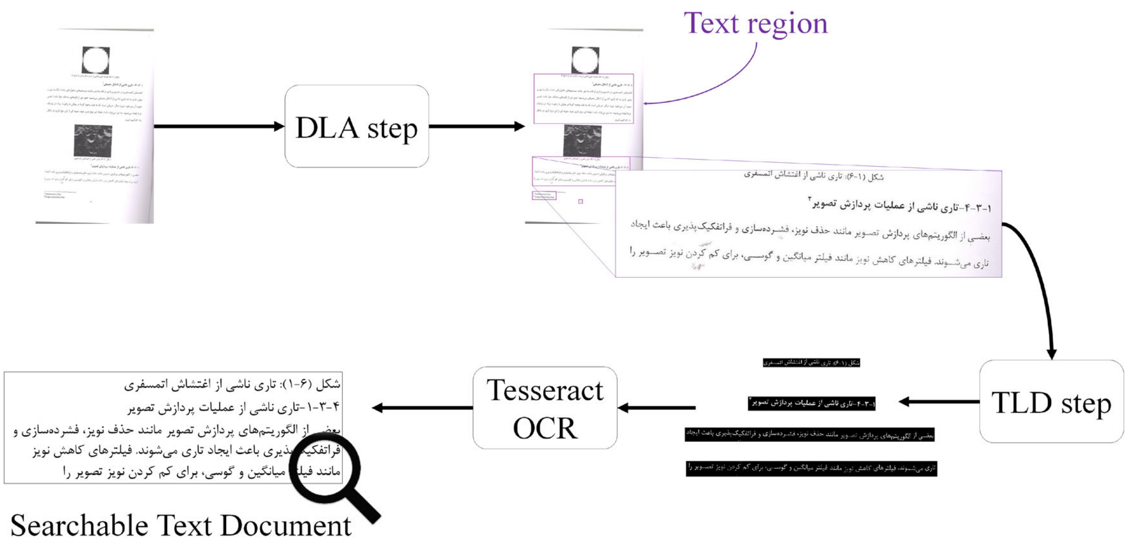
This section presents our proposed method, comprising three essential steps: DLA, TLD, and OCR, where Tesseract-OCR 5.1.0 was employed. The primary objective of the DLA stage is to divide the input image into distinct text and non-text regions. Once the DLA module has processed the image, the TLD stage takes over, identifying and segmenting the text areas into individual lines. By leveraging advanced techniques and algorithms, our method aims to extract and categorize textual information from complex documents accurately. Figure 1 illustrates the pipeline of our proposed method.

3.1 | Document layout analysis step

In this subsection, we provide a detailed description of the DLA step in our proposed method for extracting textual regions from the input image. Our approach harnesses the power of four deep learning models: YOLOv3, SSD, Faster RCNN, and Layout Parser. These models have demonstrated high accuracy and speed, making them well-suited for the task at hand.

The input image is inputted into each of the four models to initiate the DLA process, as depicted in Figure 2. Each model generates a data frame containing vital information, including the coordinates of the extracted bounding boxes and the classification of the regions as textual or non-textual. These predictions serve as valuable inputs for subsequent steps in our method.

However, what sets our DLA method apart and significantly elevates its accuracy is our innovative integration of a multi-model voting system. This ingenious approach harnesses the predictive power of each model, transcending the boundaries of individual pixel-level decisions. Instead, it introduces a sophisticated 5 × 5 window mechanism for evaluating the textual and non-textual nature of regions. The result is a dynamic voting system that intelligently aggregates



شکل (۶-۱): تاری ناشی از اغتشاش اتمسفری
۴-۳-۱-تاری ناشی از عملیات پردازش تصویر
بعضی از الگوریتم‌های پردازش تصویر مانند حذف نویز، فشرده‌سازی و
فراتفکیک‌پذیری باعث ایجاد تاری می‌شوند. فیلترهای کاهش نویز
مانند فیلتر میانگین و گوسی، برای کم کردن نویز تصویر را

FIGURE 1 Block diagram of the proposed method.

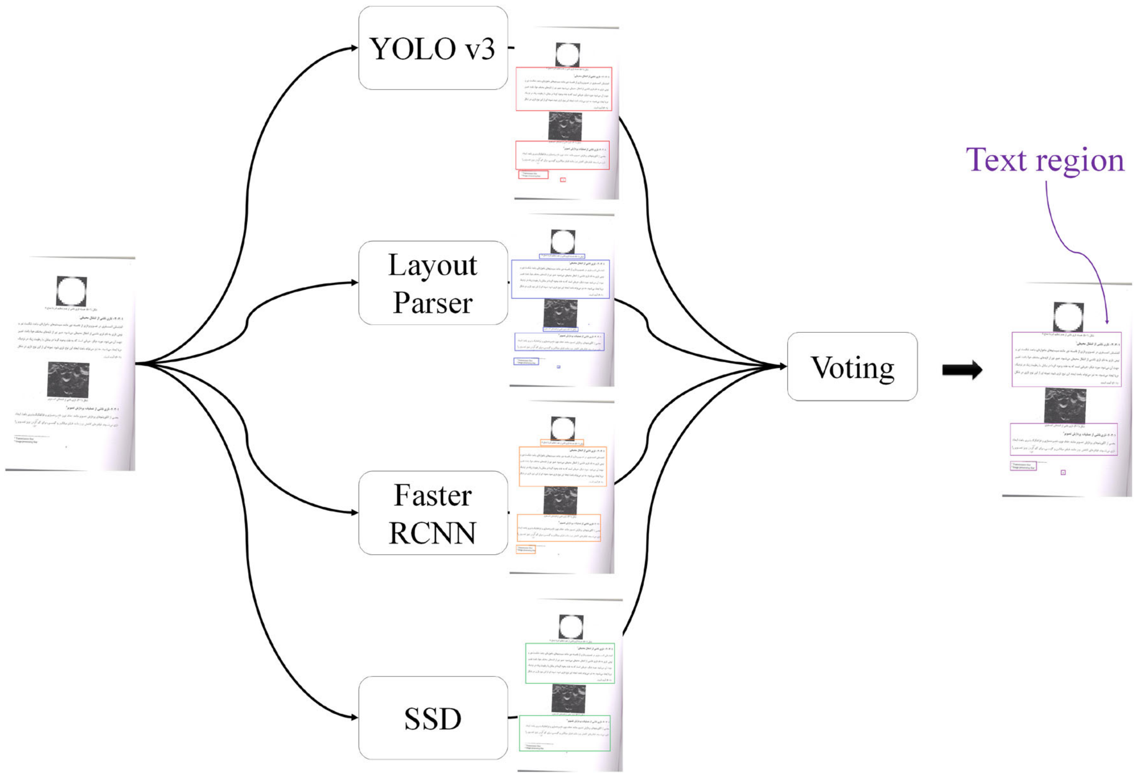


FIGURE 2 Block diagram of the document layout analysis step.

predictions from all models, reaching a consensus that capitalizes on the collective intelligence of these advanced deep learning algorithms. Regions that accumulate the highest number of votes are deemed the most probable textual and non-textual regions, playing a pivotal role in guiding the subsequent phases of our method.

In summary, our DLA step represents a paradigm shift in OCR methodology, underpinned by state-of-the-art deep learning models and a novel multi-model voting system, introduced as follows. These innovations collectively bolster the accuracy, efficiency, and robustness of our OCR system, marking a significant leap forward in the field of text extraction from diverse document layouts.

3.1.1 | Voting

In the voting subsection, we focus on determining the most probable textual and non-textual areas based on the information obtained from the text and non-text matrices derived in the previous section. These matrices contain values ranging from zero to four, representing the likelihood of a region being classified as textual or non-textual.

For each region in the image, we examine the values in the corresponding positions of the text and non-text matrices. A four-value in the non-textual matrix indicates that four methods have identified the pixel as non-textual. In contrast, a value of one in the textual matrix suggests that one method has recognized it as textual. It is important to note that the values in these matrices are bounded between zero and four.

Sometimes, the sum of textual and non-textual values may exceed four. This situation arises when different regions overlap within a single method. Such overlaps occur when a pixel is assigned to the same method's textual and non-textual matrices.

To resolve this, we employ a voting system to determine the classification of each pixel and subsequently assign the corresponding areas in the text and non-text matrices. Suppose the difference between the values of the two matrices for a given pixel is greater than or equal to one. In that case, the voting system places that pixel in the category of the region with the larger matrix value. For example, if the pixel value (x_i, y_j) in its corresponding area of the text matrix is four, while in the non-text matrix, it is one, the voting system assigns this pixel as part of the text area.

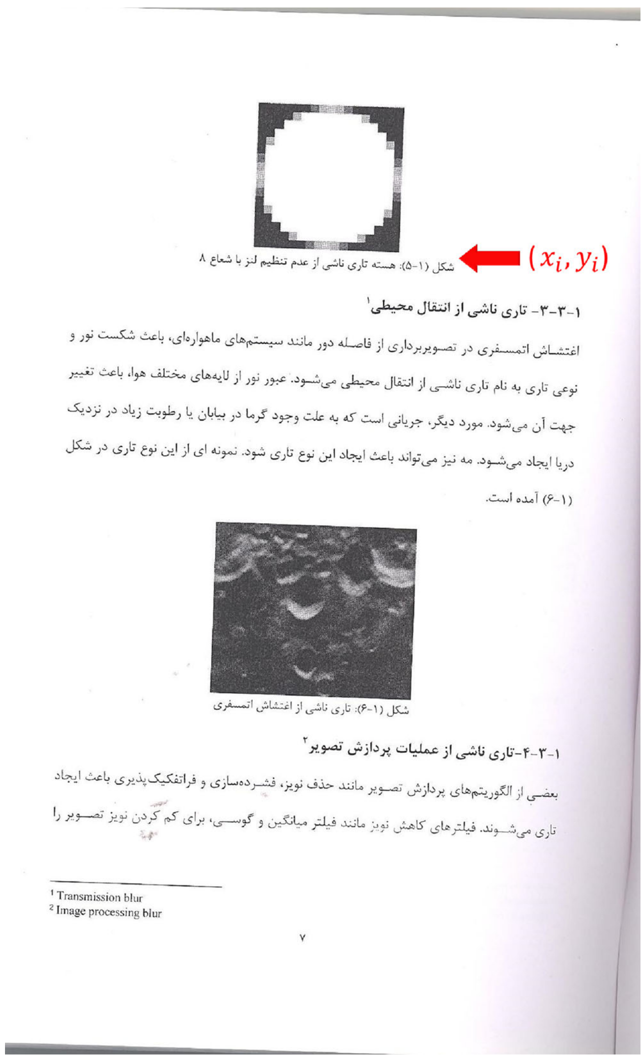
By applying this voting mechanism, we prioritize the regions with higher matrix values and make informed decisions regarding the classification of pixels and the corresponding areas in the text and non-text matrices.

This voting process ensures the identification of the most likely textual and non-textual areas within the image, considering the contributions of multiple pixel methods. The following sections delve further into evaluating and analysing our voting system.

However, the voting system sometimes encounters challenges, particularly when some pixels cannot be definitively assigned to textual or non-textual regions. These pixels are referred to as disputed pixels. To address this issue, an auxiliary method is employed to determine the assignment of these disputed pixels.

When dealing with disputed pixels, we employ a 5×5 window centered around the pixel in question to guide the voting process. Let's consider Figure 3 as an example where a disputed pixel is depicted. The voting system opens a 5×5 window centered on the disputed pixel and retrieves the values of the corresponding regions within this window from the text and non-text matrices. The prediction results of four methods for this 5×5 window are provided in Tables 1 and 2.

Upon examining the values in the window, we observe that the total number of non-textual votes is higher than the textual votes. Consequently, this disputed pixel will be classified as one of the non-textual pixels. By extending this process to all disputed pixels, we can determine the classification of all pixels in the image and effectively separate the text and non-text regions.



شکل (۱-۵): هسته تاری ناشی از عدم تنظیم لنز با شعاع ۸

۱-۳-۳- تاری ناشی از انتقال محیطی[۱]

اغتشاش اتمسفری در تصویربرداری از فاصله دور مانند سیستم‌های ماهواره‌ای، باعث شکست نور و نوعی تاری به نام تاری ناشی از انتقال محیطی می‌شود. عبور نور از لایه‌های مختلف هوا، باعث تغییر جهت آن می‌شود. مورد دیگر، جریانی است که به علت وجود گرما در بیابان یا رطوبت زیاد در نزدیک دریا ایجاد می‌شود. مه نیز می‌تواند باعث ایجاد این نوع تاری شود. نمونه ای از این نوع تاری در شکل (۱-۶) آمده است.

شکل (۱-۶): تاری ناشی از اغتشاش اتمسفری

۱-۳-۴-تاری ناشی از عملیات پردازش تصویر[۲]

بعضی از الگوریتم‌های پردازش تصویر مانند حذف نویز، فشرده‌سازی و فراتفکیک‌پذیری باعث ایجاد تاری می‌شوند. فیلترهای کاهش نویز مانند فیلتر میانگین و گوسی، برای کم کردن نویز تصویر را

[1] Transmission blur
[2] Image processing blur

۷

FIGURE 3 Sample of disputed pixel.

TABLE 1 The sum of the values of the corresponding regions of the 5 × 5 window in four non-textual matrices.

Non-text sum = 44	$j-2$	$j-1$	j	$j+1$	$j+2$
$i-2$	1	1	2	3	3
$i-1$	1	1	2	3	3
i	1	1	2	3	3
$i+1$	1	1	1	2	2
$i+2$	1	1	1	2	2

TABLE 2 The sum of the values of the corresponding regions of the 5 × 5 window in four textual matrices.

Text sum = 22	$j-2$	$j-1$	j	$j+1$	$j+2$
$i-2$	1	1	0	0	0
$i-1$	2	2	2	1	1
i	2	2	2	1	1
$i+1$	2	2	2	1	1
$i+2$	2	2	2	1	1

By utilizing the 5 × 5 window and examining corresponding regions within it, we ensure a more accurate determination of the classification for disputed pixels. This approach enhances the overall reliability of our voting system and facilitates the comprehensive separation of the text and non-text portions within the image.

To ensure the utmost accuracy and robustness in the DLA step, we understand the paramount importance of ongoing refinement of our foundational techniques. Although our initial implementation harnesses the strengths of four deep learning models—YOLOv3, SSD, faster RCNN, and Layout Parser—for the precise identification of textual and non-textual regions, our dedication to research and development remains unwavering. These continual improvements serve as a cornerstone for our proposed OCR system's effectiveness, directly influencing the quality of information supplied to the voting mechanism. Through the enhancement of these foundational techniques, we aspire to attain even higher accuracy in the identification of textual and non-textual regions, thereby fortifying the bedrock upon which our voting system operates.

In the following sections, we will delve deeper into the evaluation and validation of our voting system, providing a comprehensive assessment of its effectiveness and performance. Moreover, we acknowledge the vital role of continuous refinement in our foundational techniques as an indispensable component in enhancing the overall capabilities of our OCR system. Our relentless pursuit of progress in both the voting mechanism and core methods underscores our unwavering commitment to delivering state-of-the-art OCR technology.

3.2 | Text line detection step

To elevate the TLD step, our proposed method incorporates a series of groundbreaking techniques, each playing a vital role in ensuring the precise extraction of text lines from scanned OIN images. Illustrated in Figure 4, the TLD process encompasses several key sub-processes: preprocessing, recognition of the optimum font size, elimination of dots and diacritics, angle correction, line extraction, and the elimination of line curvature. In the initial preprocessing step, we meticulously segregate text blocks from non-text blocks, followed by denoising and binary conversion to prepare the text blocks for further analysis. Our method introduces an innovative concept known as the "optimum font size." By leveraging this novel approach, we effectively determine the optimum font size for each text block. This determination is instrumental in the subsequent elimination of dots and diacritics, addressing issues that might impede the angle correction process. Angle detection, a critical phase in our approach, relies on precise measurement of the angle between the baseline and the horizontal line within each text block. Once the baseline is identified, the angle is calculated, and the image is subsequently rotated to eliminate skew. This critical information guides the process of connecting connected components (CCs) within each line, facilitating the precise extraction of text line areas. Positional information for each text line is

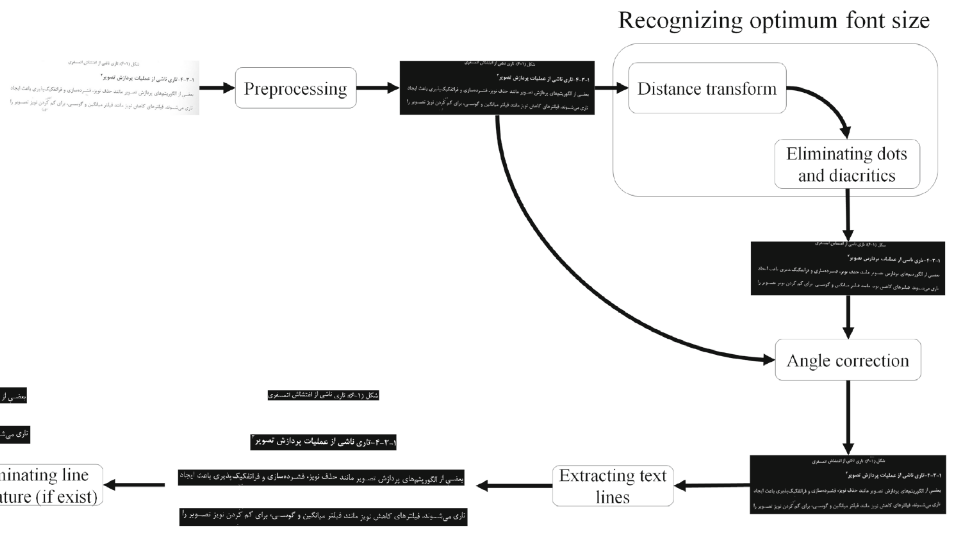

Recognizing optimum font size
Preprocessing
Distance transform
Eliminating dots and diacritics
Angle correction
Extracting text lines
Eliminating line curvature (if exist)

FIGURE 4 Block diagram of the proposed method model.

derived through bounding box analysis, a key step in our method. In cases where a bounding box encompasses multiple lines, our method excels in accurately separating these lines, further enhancing the precision of text line extraction. Furthermore, our method addresses a critical challenge in text line extraction—line curvature. We accomplish this by strategically aligning words with the baseline, effectively eliminating curvature issues. These sophisticated techniques collectively contribute to our method's unparalleled accuracy in text line extraction, all while mitigating the challenges posed by the scarcity of labeled Persian data for deep learning-based approaches.

3.2.1 | Preprocessing

Preprocessing is the first step in extracting text lines. As said before, the proposed method needs to extract text blocks from the background. For this purpose, we used our previous work, which is a voting-based method from four different deep neural networks, for layout analysis.[6] These four deep neural networks are Faster-RCNN,[14] YOLOv3,[17] SSD,[15] and Layout-Parser.[16] Due to the better performance of the voting-based method than these four methods, we have used it to extract text blocks. After extracting text blocks, as shown in Algorithm 1, the proposed method converts the RGB input image to grayscale by eliminating the hue and saturation information while retaining the luminance (Figure 5A). Then, it eliminates image noises by using pre-trained denoising deep neural network[38] (Figure 5B). To increase the performance of the binary function, and given that, in text images, the foreground is darker than the background, adaptive thresholding, called Bradley's method, is used to convert the grayscale image to a binary image (Figure 5C).[39] Finally, the binarization is performed by setting these values to 1 and the rest of the pixels (background) to zero (Figure 5D).

Algorithm 1. Preprocessing steps

```
1: procedure DENOISING-BINARY($Input - image$)
2:     grayscale-image ← rgb2gray($Input - image$)
3:     modified-grayscale-image ← PretrainedCNN(grayscale-image)
4:     binary-image ← AdaptiveBinarization(modified-grayscale-image)
5:     binary-image ← not(binary-image)
6:     Return modified-binary-image
7: end procedure
```

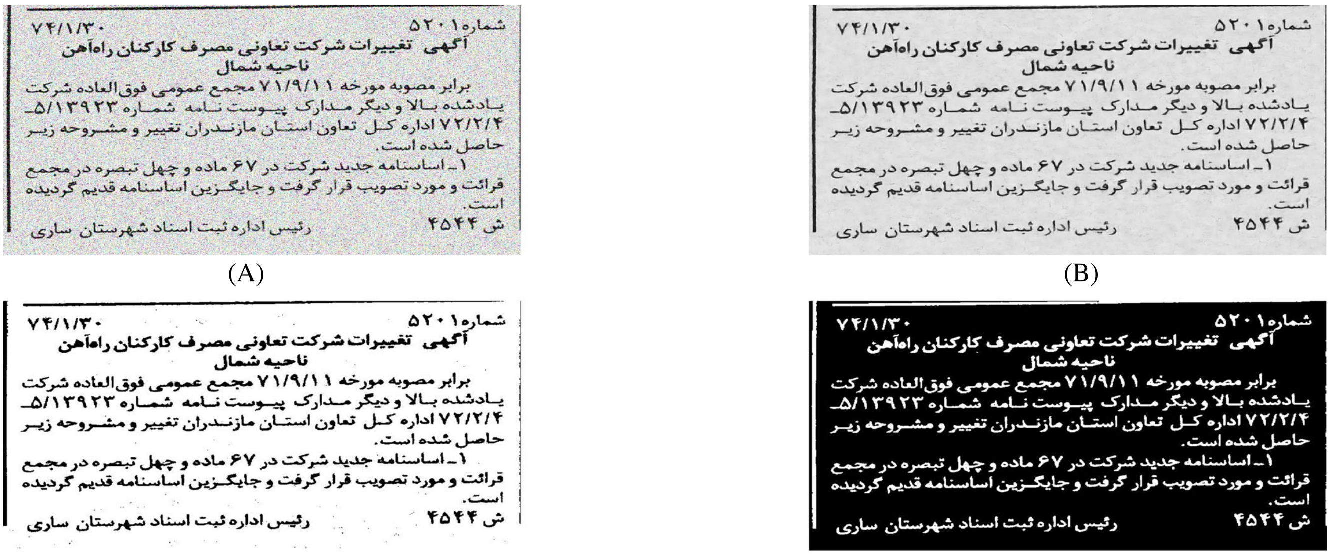

شماره ۵۲۰۱ ۷۴/۱/۳۰
آگهی تغییرات شرکت تعاونی مصرف کارکنان راه‌آهن
ناحیه شمال
برابر مصوبه مورخه ۷۱/۹/۱۱ مجمع عمومی فوق العاده شرکت یادشده بالا و دیگر مدارک پیوست نامه شماره ۵/۱۳۹۲۳ـ ۷۲/۲/۴ اداره کل تعاون استان مازندران تغییر و مشروحه زیر حاصل شده است.
۱ـ اساسنامه جدید شرکت در ۶۷ ماده و چهل تبصره در مجمع قرائت و مورد تصویب قرار گرفت و جایگزین اساسنامه قدیم گردیده است.
ش ۴۵۴۴ رئیس اداره ثبت اسناد شهرستان ساری

FIGURE 5 Different steps of preprocessing. (A) Preprocessing input. (B) De-noised image. (C) Binary image. (D) Preprocessing output.

3.2.2 | Recognizing optimum font size

This subsection is divided into two parts. First, we describe how the proposed method is used to calculate the diameter of the CCs in the image (Algorithm 2). Then, the optimum font size of the image is extracted according to the diameter of the CCs (Algorithm 3). By extracting the optimum font size of the image, we can separate the lines well. We also prevent curved lines from connecting.

In Algorithm 2, we leverage the power of distance transform to accurately determine the diameter of each connected component (CC) per pixel. This crucial step plays a pivotal role in recognizing the optimum font size, as its precision directly influences text line extraction. As elucidated in Figure 6, the distance transform assigns a numeric value to each pixel in the image, reflecting its proximity to the nearest nonzero pixel. To initiate this process, we first negate the binary image, effectively converting the background to white. The Euclidean distance metric is employed to calculate these distances accurately.

Subsequently, we meticulously analyze the results of the distance transform to identify the diameter with the most significant number of repetitions within each connected component. This carefully selected diameter is deemed the

Algorithm 2. CC's diameter calculation steps

```
1: Input: binary-image, modified-binary-image
2: Output: CCs' diameter
3: binary-image ← not(modified-binary-image)
4: for $i = 1$ to Number of CCs do
5:     $CC_i$'s diameter in each pixel ← distance-transform(binary-image)
6: end for
7: for $i = 1$ to Number of CCs do
8:     optimum $CC_i$'s diameter ← the largest number of repetitions of the diameter among the $CC_i$
9:     for each pixel of $CC_i$ do
10:        $CC_i$'s diameter of the pixel ← optimum $CC_i$'s diameter
11:    end for
12: end for
```

Algorithm 3. Optimum font size recognition steps

```
Require: $CC_i$'s pixels for $i = 1, \dots,$ Number of CCs
Ensure: optimum font size
1: CandidateFontSize ← Diameter with the most repetitions
2: for $i = 1$ to Number of CCs do
3:     if Number of $CC_i$'s pixels $< 10 \times$ CandidateFontSize then
4:         Remove $CC_i$ ($CC_i$'s Label = background Label)
5:         Remove $CC_i$'s diameters from the diameter's table
6:     end if
7: end for
8: UD ← Sort unique diameters
9: UDNum ← Calculate the number of repetitions of unique diameters
10: OptimumFontSize ← Diameter with the most repetitions
11: Return OptimumFontSize
```

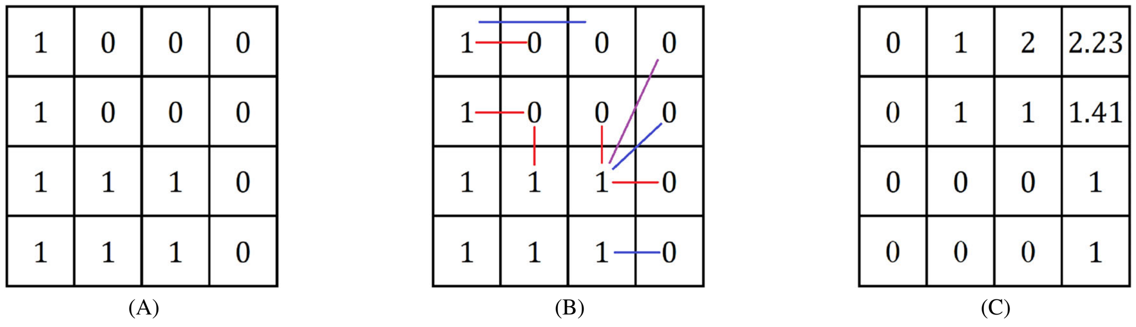


FIGURE 6 Distance transform steps. (A) Input image. (B) Finding nearest nonzero pixel. (C) Output of distance transform.

optimum diameter for the respective CC. Importantly, all pixels within a given connected component are attributed this optimum diameter, a critical factor in ensuring precise and consistent font size recognition across the image, contributing significantly to the overall accuracy of the text line extraction process.

In the next step, the proposed method takes all CC's diameters with their repetitions. Among these diameter values, the one with the most significant number of repetitions is selected and labeled as the initial font size candidate. Also, all the dots and diacritics are removed to improve this step's performance before calculating repetitions. Otherwise, the proposed method may mistakenly calculate the optimum font size based on the diameter of these dots and diacritics, which is wrong. For further clarification, Figure 7 shows some examples of different kinds of diacritic in Persian.

To remove all diacritics and dots, the number of CC's pixels needs to be checked. If the number of each CC's pixels is less than a predetermined value, the CC is removed. This means that the CC is so minor that it cannot be considered a full letter. By doing this, all small objects are removed from binary images. Therefore, the most significant number of repetitions of the diameter in all CCs is found. This is followed by calculating the number of pixels of each CC. Finally, the proposed method removes CCs whose pixels are less than ten times the diameter with the most repetition. The ten times factor has been achieved after many trials and errors on different font types in various sizes and styles. The corresponding CC's diameters are removed from the diameters table by removing a CC. After eliminating all dots and diacritics, the remaining diameters on the image are sorted. Then, the diameter with the most significant number of repetitions is chosen as the optimum font size.

3.2.3 | Angle correction

Angle correction is a pivotal component of the TLD method, addressing the challenge posed by skewed images. Our proposed method employs an advanced angle correction technique to rectify image skew, as illustrated in Figure 8A. In

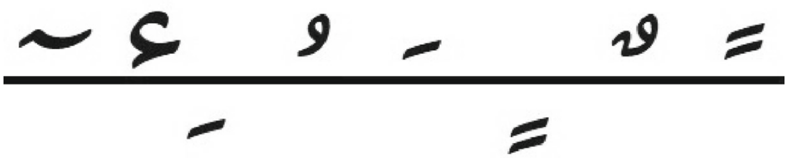
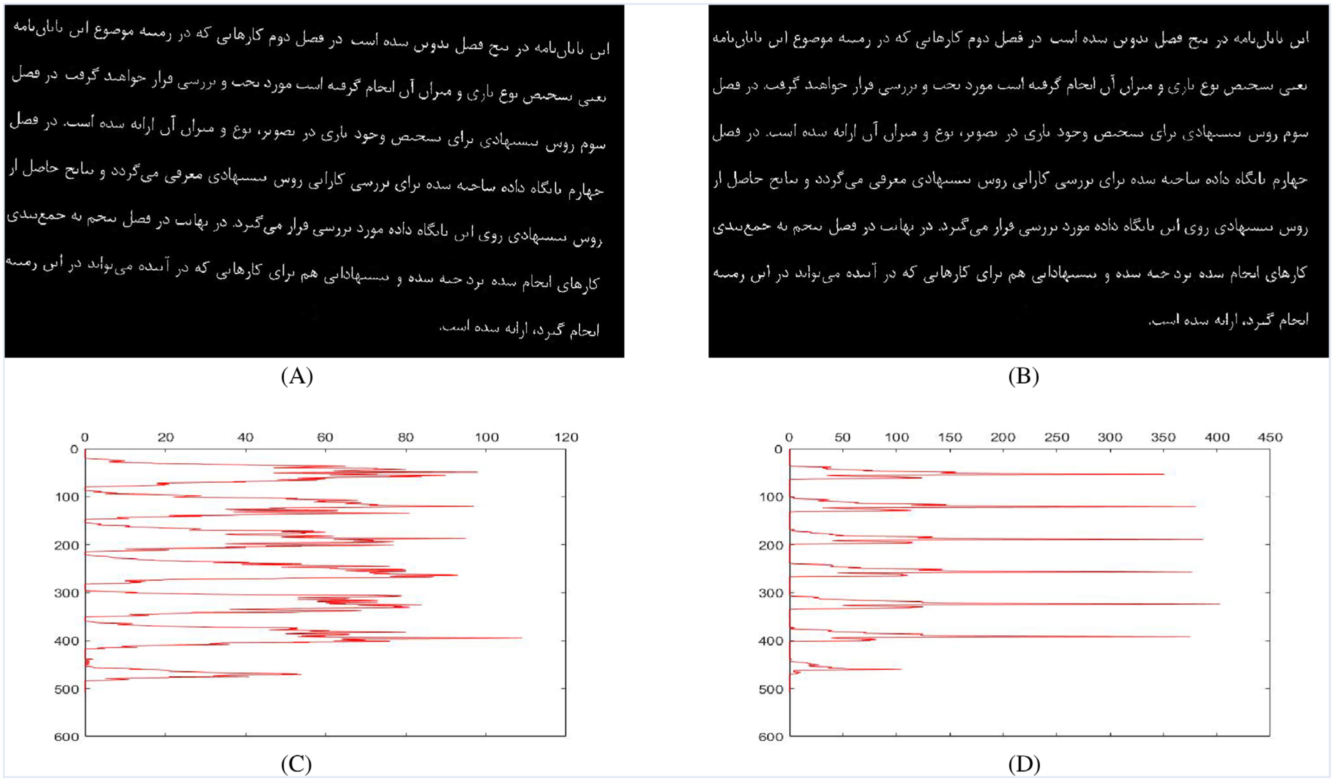


FIGURE 7 Different kinds of diacritic in Persian.



اس ىاىان‌نامه در ىىح فصل ىدوىن سده است در فصل دوم کارهاىى که در زمىنه موضوع اىن ىاىان‌نامه
ىعنى ىسحىص ىوع ىارى و مىزان آن انحام گرفىه است مورد ىحث و ىررسى قرار حواهد گرفت. در فصل
سوم روس ىىسىهادى ىراى ىسحىص وحود ىارى در ىصوىر، ىوع و مىزان آن ارائه سده است. در فصل
چهارم ىاىگاه داده ساحىه سده ىراى ىررسى کاراىى روس ىىسىهادى معرفى مى‌گردد و ىىاىح حاصل از
روس ىىسىهادى روى اىن ىاىگاه داده مورد ىررسى قرار مى‌گىرد. در ىهاىت در فصل ىىحم ىه حمع‌ىندى
کارهاى انحام سده ىرداحىه سده و ىىسىهاداىى هم ىراى کارهاىى که در آىنده مى‌ىواند در اىن زمىنه
انحام گىرد، ارائه سده است.

(A)

اس ىاىان‌نامه در ىىح فصل ىدوىن سده است در فصل دوم کارهاىى که در زمىنه موضوع اىن ىاىان‌نامه
ىعنى ىسحىص ىوع ىارى و مىزان آن انحام گرفىه است مورد ىحث و ىررسى قرار حواهد گرفت. در فصل
سوم روس ىىسىهادى ىراى ىسحىص وحود ىارى در ىصوىر، ىوع و مىزان آن ارائه سده است. در فصل
چهارم ىاىگاه داده ساحىه سده ىراى ىررسى کاراىى روس ىىسىهادى معرفى مى‌گردد و ىىاىح حاصل از
روس ىىسىهادى روى اىن ىاىگاه داده مورد ىررسى قرار مى‌گىرد. در ىهاىت در فصل ىىحم ىه حمع‌ىندى
کارهاى انحام سده ىرداحىه سده و ىىسىهاداىى هم ىراى کارهاىى که در آىنده مى‌ىواند در اىن زمىنه
انحام گىرد، ارائه سده است.

(B)

FIGURE 8 Angle correction procedure. (A) Skewed image. (B) Modified image. (C) Horizontal projection of skewed image. (D) Horizontal projection of modified image.

some images, the angle of lines can be quite pronounced, significantly impacting the accuracy of line detection algorithms. Figure 8C demonstrates that the horizontal projection yields a more substantial number of rows with non-zero values when the image is skewed. To remedy this, we adopt a systematic approach, rotating the image at various angles to determine the optimum degree of rotation. The degree at which the image's horizontal projection exhibits the maximum number of rows with values close to zero is deemed the most suitable.

As shown in Algorithm 4, the image is rotated for *i* degree. To increase the speed of the proposed method and by analysing the images of the dataset, we considered 10° as the step-size of the rotation angle. Hence, the proposed method rotates the image clockwise by *i* degree and then calculates the projection of the rotated image. In this step, the more rows obtained close to zero in the image projection, the more appropriate angle obtained is.

The angle correction process is reiterated multiple times to pinpoint the optimum degree and fine-tune the results. In this refined phase, the range of angle adjustments is limited to 2°, centered on the best degree, with rotation increments of just 0.1°, a significant enhancement over the initial 1° increments. As a result, the method attains the best rotation degree. After determining the optimum angle, we apply it to the original image, which may contain dots and diacritics. Figure 8 visually demonstrates the effect of this process: post-correction, the lines are non-skewed, and we achieve the most accurate projection, significantly improving line detection accuracy.

Algorithm 4. Angle correction steps

```
procedure PROJECTION(binary-image)
    $W \leftarrow$ binary-image's width
    projection $\leftarrow$ Number of white pixels per row in binary-image
    $x \leftarrow 0$
    for $i \leftarrow 1$ to len(projection) step 1 do
        if projection[$i$] $\leq 0.01 \cdot W$ then
            $x \leftarrow x + 1$
        end if
    end for
    Return $x$
end procedure
procedure MAIN(binary-image)
    best-degree $\leftarrow$ none
    $x \leftarrow 0$
    for $i \leftarrow -10$ to 10 step 1 do
        rotated-image $\leftarrow$ imrotate(binary-image, $i$)
        $x_{new} \leftarrow$ projection(rotated-image)
        if $x < x_{new}$ then
            $x \leftarrow x_{new}$
            best-degree $\leftarrow i$
        end if
    end for
    Fbest-degree $\leftarrow$ best-degree
    for $i \leftarrow$ Fbest-degree - 1 to Fbest-degree + 1 step 0.1 do
        rotated-image $\leftarrow$ imrotate(binary-image, $i$)
        $x_{new} \leftarrow$ projection(rotated-image)
        if $x < x_{new}$ then
            $x \leftarrow x_{new}$
            best-degree $\leftarrow i$
        end if
    end for
    non-skewed-image $\leftarrow$ imrotate(binary-image, best-degree)
    Return non-skewed-image
end procedure
```

3.2.4 | Extracting text lines

To determine the approximate lines in the image, the proposed method involves connecting every connected component (CC) horizontally in the same direction. Additionally, each CC searches for its closest horizontal neighbor using a search method. The optimum font size and search length are used as criteria to find another CC, and if any white pixel is found during the search process, the two CCs are connected. The search length is controlled by an adaptive coefficient, which is determined by multiplying it with the optimum font size to obtain the best search length.

Then, morphological structuring elements called *strel* is employed. Its basic syntax is:

$$se = strel(Shape, parameters), \tag{1}$$

which the proposed method used one of the special types:

$$se = strel('line,' Len, Deg). \tag{2}$$

The proposed method employs a flat linear structuring element, referred to as "strel," where the "Len" parameter specifies the length and "Deg" parameter defines the angle of the line in degrees. By utilizing the morphological operation of "closing," the method connects connected components (CCs) within a specific search radius, enabling the accurate detection of lines. Algorithm 5 illustrates how to calculate the adaptive coefficient.

The study faced a significant challenge in determining the optimum search length, particularly when dealing with images containing curved text lines (as seen in Figure 9A). As demonstrated in Figure 9B, in some instances, the search algorithm cannot locate any CCs near the primary CC due to the short search length, resulting in an incorrect line detection. Conversely, as depicted in Figure 9C, increasing the search length may lead to the proposed method's inability to correctly separate two consecutive lines in a column and connect them. The proposed adaptive coefficient, which eliminates the need to detect curve lines, is obtained by multiplying the optimum font size by this coefficient (as shown in Figure 9D). This approach overcomes the challenges and inaccuracies encountered when the search length is either increased or decreased.

Although the proposed method correctly executes all the preceding steps to divide the lines, it detects some points of the word above and below as a new label, as illustrated in Figure 10A. To address this issue, the proposed method includes a supplementary step at this stage. Initially, the method encloses each label in a box and identifies coordinate positions (x_1, y_1, x_2, and y_2) for every label, which we term min(x), max(x), min(y), and max(y). As shown in Figure 10B, if the position of the label for the dot or diacritic is within the bounding box of the primary label, the method connects the two labels. If the label for the dot is outside the bounding box, it will be regarded as an independent connected component (CC). If a diacritic's label's position is not within any bounding box, the diacritic is connected to the bottom line.

Algorithm 5. Finding the best searching length steps

```
Require: optimum-font-size
Ensure: main body of each line
1: $A \leftarrow 1$
2: if optimum-font-size<50 then
3:     $A \leftarrow \frac{60-\text{optimum-font-size}}{10}$
4: end if
5: searching-length ← $A \times$ optimum-font-size
6: se ← strel('line', searching-length, 0)
7: image ← imclose(image, se)
```

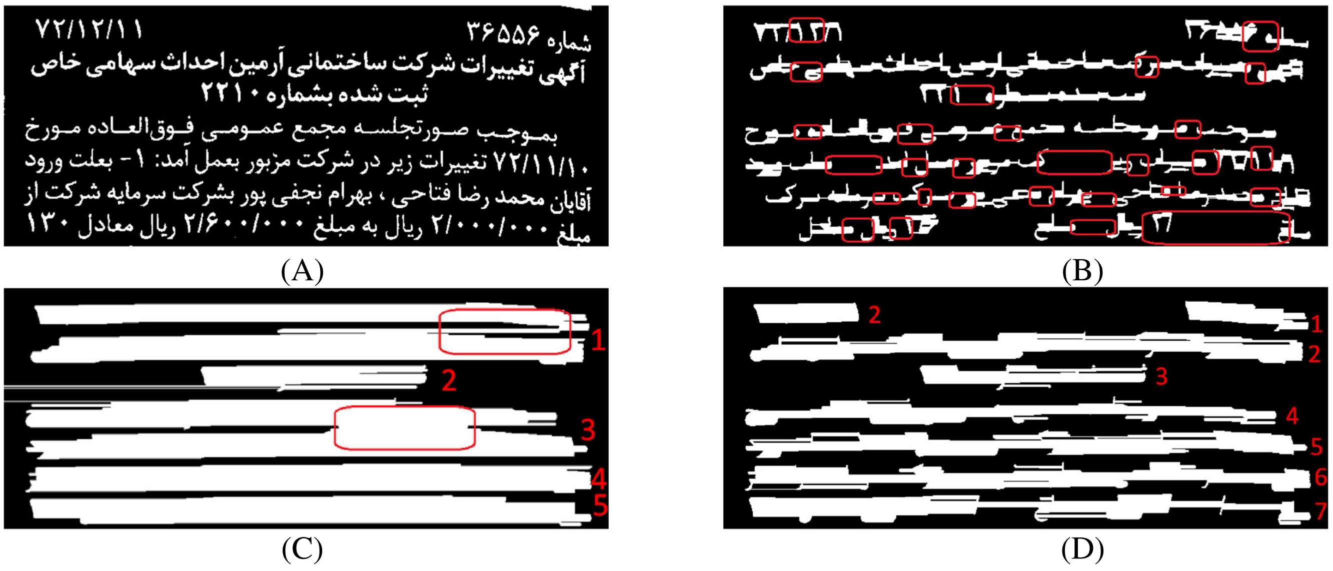

شماره ۳۶۵۵۶ ۷۲/۱۲/۱۱
آگهی تغییرات شرکت ساختمانی آرمین احداث سهامی خاص
ثبت شده بشماره ۲۲۱۰
بموجب صورتجلسه مجمع عمومی فوق‌العاده مورخ
۷۲/۱۱/۱۰ تغییرات زیر در شرکت مزبور بعمل آمد: ۱- بعلت ورود
آقایان محمد رضا فتاحی ، بهرام نجفی پور بشرکت سرمایه شرکت از
مبلغ ۲/۰۰۰/۰۰۰ ریال به مبلغ ۲/۶۰۰/۰۰۰ ریال معادل ۱۳۰

FIGURE 9 Searching length problem. (A) Preprocessing input. (B) Short searching length. (C) Long searching length. (D) Best searching length.

FIGURE 10 Bounding box steps. (A) Disconnected dots. (B) Bounding box.

In addition to all the problems mentioned earlier and the proposed solutions, it is still possible for two or even more than two lines to be connected. The dataset presented in this article has many complexities that may reduce the accuracy of line extraction systems. For example, if there is diacritic on some letters of a line, the distance between this line and the upper line becomes less than usual; as a result, two lines or even more may be connected. As shown in Algorithm 6, the line height is measured after extracting lines and using the bounding box to resolve this issue.

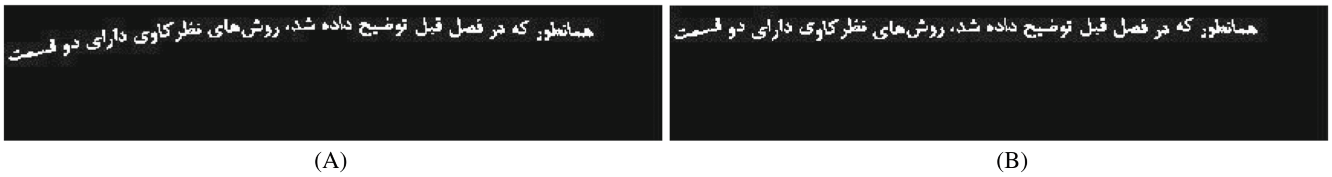
If the line height is more than twice the median value of lines height, the algorithm detects that two lines have been connected. This process is achieved using the line's horizontal histogram. The dot near the center of the line with the lowest pixel count is considered the boundary separating the two lines. Similarly, if more than two lines are connected, the proposed method separates one of the lines in each step.

3.2.5 | Eliminating line curvature

In this subsection, which is the last step of the proposed method, the curvature of the line is primarily eliminated, and a line without curvature would become ready to be fed to the OCR system. As shown in Figure 11A most of the lines in the OIN dataset are curved. If there is a curvature in the line, OCR systems have to eliminate the curvature, which is sometimes tricky. As shown in Algorithm 7, the proposed method eliminates line curvature by presenting a new approach.

Algorithm 6. Separating two connected lines

```
Input: Number of lines, $x, y$
Output: Separated lines
for $i = 1$ to Number of lines do
    $height[i] \leftarrow \max(y[i]) - \min(y[i])$
end for
$median\_height \leftarrow$ Median($height[:]$)
for $i = 1$ to Number of lines do
    while $height[i] > 2 \times median\_height$ do
        $projection \leftarrow$ Number of white pixels per row in bounding box of line
        $center \leftarrow \min(y[i]) + \frac{height[i]}{2}$
        $[\_, index] \leftarrow \min(projection[center - \frac{height[i]}{4} : center + \frac{height[i]}{4}])$
        line's Separator boundary ← index
        # Adding a new line to the bottom of the lines list
        Number of lines ← Number of lines +1
        $\max(y[Numberoflines]) \leftarrow \max(y[i])$
        $\min(y[Numberoflines]) \leftarrow index + 1$
        $\max(x[Numberoflines]) \leftarrow \max(x[i])$
        $\min(x[Numberoflines]) \leftarrow \min(x[i])$
        $height[Numberoflines] \leftarrow \max(y[Numberoflines]) - index - 1$
        # Updating Informations of line[i]
        $\max(y[i]) \leftarrow$ index
        $height[i] \leftarrow index - \min(y[i])$
    end while
end for
```



همانطور که در فصل قبل توضیح داده شد، روش‌های نظرکاوی دارای دو قسمت

همانطور که در فصل قبل توضیح داده شد، روش‌های نظرکاوی دارای دو قسمت

FIGURE 11 Eliminating line curvature steps. (A) Curved line. (B) Non curved line.

To eliminate the curvature of the line, we need to extract the baseline and then eliminate the line's curvature by placing the words to the baseline. In languages such as Persian, Arabic, and Urdu, most of a word's characters in the baseline are exactly connected together. Therefore, the baseline has the most significant number of white pixels. Thus, the baseline can be extracted using projection. But as shown in Algorithm 7, after some trial and error, the proposed method separates 20% of the initial columns and 20% of the last columns from the curved line due to the significant curvature at the beginning and end of the line. By separating these areas, the probability of extracting the real baseline will increase.

After extracting the baseline, the subwords need to be extracted. Therefore, the proposed method uses Algorithms 2 and 3 to choose the optimum font size of the image. The proposed method considers twice the optimum font size as the minimum length of a subword. Next, the proposed method calculates the vertical projection of the image. The distances between two non-adjacent dots whose vertical projection values are zero indicate a subword. If the length of a subword is less than the minimum length considered by the proposed method, the subword will be combined with the nearest subword because subwords with a small length may not have a baseline.

At the end of this step, the proposed method extracts the baseline relevant to each subword. As shown in Figure 11B subwords whose baselines are spaced from the main baseline are shifted by the distance between the two baselines to significantly eliminate the curvature of the line. Also, if the baseline of a subword is not between that of its left and right subwords, the baseline coordinates will be corrected.

In summary, the meticulous process of eliminating line curvature in our method is pivotal for enhancing the accuracy and effectiveness of OCR systems, especially when combined with Tesseract-OCR. By extracting and rectifying the baselines associated with each subword, we ensure that subwords are precisely aligned with the main baseline, effectively eliminating unwanted curvature and distortion in the text lines. This critical step not only contributes to improving the visual quality of recognized text but also facilitates smoother integration with subsequent OCR engines. The result is a refined and robust OCR solution that excels in handling challenging documents with curved or skewed text lines, further underscoring the effectiveness and innovation of our proposed approach.

3.3 | Tesseract-OCR

In this subsection, we provide an overview of Tesseract-OCR integration into our proposed method, specifically after the TLD step. Tesseract-OCR is a widely-used OCR engine known for its accuracy and versatility in extracting textual information from images.

Once the TLD stage in our method identifies and segments the text areas into individual lines, we utilize Tesseract-OCR 5.1.0 to perform the OCR process. Tesseract-OCR processes the extracted text lines and applies advanced algorithms and machine-learning techniques to recognize and convert the textual content into machine-readable formats.

Tesseract-OCR leverages a combination of character recognition models, language data, and post-processing techniques to achieve accurate and reliable results. The engine employs statistical analysis and contextual information to enhance character recognition and handle fonts, sizes, and styles variations.

By integrating Tesseract-OCR into our proposed method, we harness its robust OCR capabilities to convert the extracted text lines into meaningful and interpretable textual information. This final stage completes the OCR pipeline, enabling textual content extraction from the input images with high accuracy and efficiency.

Algorithm 7. Eliminating line curvature

```
1:  procedure BASELINE($curved\_line$)
2:      $middle\_columns \leftarrow$ Separating 20% of the initial columns and 20% of the last columns from $curved\_line$
3:      $projection \leftarrow$ Number of white pixels per row in $middle\_columns$
4:      $[\_, index] \leftarrow \max(projection[:])$
5:      $baseline \leftarrow index$
6:      return $baseline$
7:  end procedure
8:  procedure BASELINEFORWORDS($sub\_word$)
9:      $projection \leftarrow$ Number of white pixels per row in $sub\_word$
10:     $[\_, index] \leftarrow \max(projection[:])$
11:     $baseline \leftarrow index$
12:     return $baseline$
13: end procedure
14: procedure SEPARATINGWORDS($curved\_line$)
15:     choosing optimum-font-size using Algorithms 2 and 3
16:     minimum-length $\leftarrow$ 2× optimum-font-size
17:     $vertical\_projection \leftarrow$ Number of white pixels per column in $curved\_line$
18:     $m \leftarrow 0$
19:     for $i \leftarrow 1$ to len($vertical\_projection$) do
20:         if $vertical\_projection[i] = 0$ then
21:             $zero\_point[m] \leftarrow i$
22:             $m \leftarrow m + 1$
23:         end if
24:     end for
25:     $m \leftarrow 0$
26:     for $i \leftarrow 1$ to Number of $zero\_point - 1$ do
27:         if $zero\_point[i+1] - zero\_point[i] > 1$ then
28:             adding a new sub_word
29:             $len\_sub\_word[m] \leftarrow$
30:                 $zero\_point[i+1] - zero\_point[i]$
31:             $m \leftarrow m + 1$
32:         end if
33:     end for
34:     for $i \leftarrow 1$ to Number of $sub\_word$ do
35:         if $len\_sub\_word[i]$ < minimum-length then
36:             finding the nearest sub_word to the sub_word[i], which is sub_word[i+1] or sub_word[i-1]
37:             $len\_sub\_word[i+1]$ or $len\_sub\_word[i-1] \leftarrow len\_sub\_word[i] +$ distance between these two sub_word
38:             removing $sub\_word[i]$
39:         end if
40:     end for
41:     return all sub_words
42: end procedure
43: procedure MAIN($curved\_line$)
44:     $baseline \leftarrow$ baseline($curved\_line$)
45:     $sub\_words \leftarrow$ SeparatingWords($curved\_line$)
46:     for $i \leftarrow 1$ to Number of $sub\_word$ do
47:         $baseline\_sub\_word[i] \leftarrow$
48:                 baselineForWords($sub\_word[i]$)
49:     end for
50:     for $i \leftarrow 2$ to Number of $sub\_word - 1$ do
51:         if $\neg(baseline\_sub\_word[i]$ is between $baseline\_sub\_word[i-1]$ and $baseline\_sub\_word[i+1])$ then
52:             $baseline\_sub\_word[i] \leftarrow$
53:             $\frac{baseline\_sub\_word[i-1] + baseline\_sub\_word[i+1]}{2}$
54:         end if
55:     end for
56:     for $i \leftarrow 1$ to Number of $sub\_word$ do
57:         $d \leftarrow baseline\_sub\_word[i] - baseline$
58:         Moving all the pixels of $sub\_word[i]$ in the amount of $-d$ pixels
59:     end for
60:     return $new\_line$
61: end procedure
```

4 | EXPERIMENTAL RESULTS

To verify the significance of the proposed method, we have performed comprehensive experiments with three different datasets for TLD step. We have used a system with 24 GB RAM and an Intel(R)-Core (TM) i7-9750H processor to implement and evaluate the proposed method. We used Matlab 2018b to implement the proposed method. With regard to the DLA step, we used Python programming within the proposed method. In what follows, we present the details of the dataset. Then, we present the results of the proposed method and system performance metrics. Following, we present the details of the dataset. Then, we present the results of the proposed method and system performance metrics.

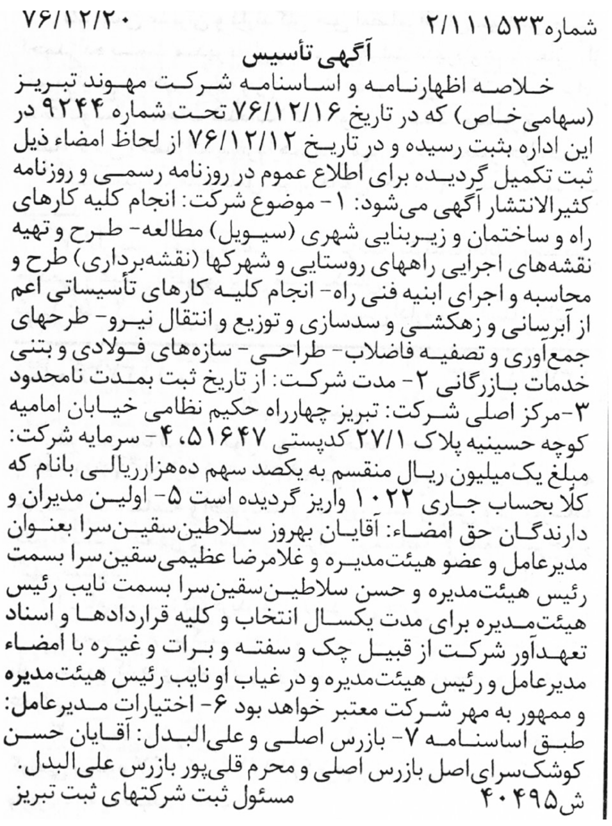
4.1 | Datasets

To address the absence of an appropriate Persian dataset for DLA and TLD, we created a new dataset consisting of 1920 images from a specific newspaper called OIN. These images were obtained using an HP Scanjet 4890 scanner with default settings. The dataset introduces complexities not found in other datasets, including skewed images, curved text lines, closely spaced lines, and numerous dots and diacritics. One notable challenge in this dataset is the presence of rotated pages, resulting in skewed lines. These challenges in the document images contribute to higher system error rates, connected lines, and decreased system speed. The images in this dataset have a resolution of 300 dpi, with a minimum width of 42 pixels, minimum height of 434 pixels, maximum width of 3846 pixels, and maximum height of 2715 pixels. For a sample of OIN dataset's images, refer to Figure 12.

In addition, to comprehensively evaluate the proposed method in the DLA step, we incorporated the PRImA dataset.[18] This dataset comprises images extracted from magazines and articles, totaling 478 images. The images within this dataset exhibit diverse shapes and contain various types of text. Unlike the dataset proposed in this article, the images in the PRImA dataset do not possess curvature or skewness, making it comparatively less complex. The PRImA dataset has been widely employed for evaluating different methods in the field of DLA. By utilizing this established dataset, we aim to benchmark the performance of our proposed method against existing state-of-the-art approaches, allowing for a comprehensive assessment of its effectiveness and robustness in real-world scenarios.

Furthermore, to further assess the performance of the proposed method in the TLD step, we employed two additional datasets for testing purposes. The second dataset, referred to as the Arabic OCR dataset, can be accessed at Reference 40. It consists of images with varying sizes, and an example is shown in Figure 13. To evaluate the method's performance, we selected the initial 50 images from this dataset and subjected them to the proposed method. While the images in this dataset mainly consist of single-column lines, they exhibit a slight rotation, adding an additional level of complexity.

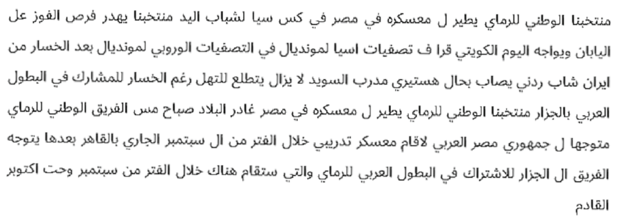
Additionally, we utilized an artificial dataset generated in a previous work.[41] This synthetic dataset was created by employing six popular Persian font types (*Tahoma, XB-Niloofar, Ziba, IranNastaliq, Arial, and Nazanin*) with regular, italic, and bold styles, along with five different font sizes (22, 18, 14, 10, and 8). The python-docx library was used to create


شماره۲/۱۱۱۵۳۳ ۷۶/۱۲/۲۰

آگهی تأسیس

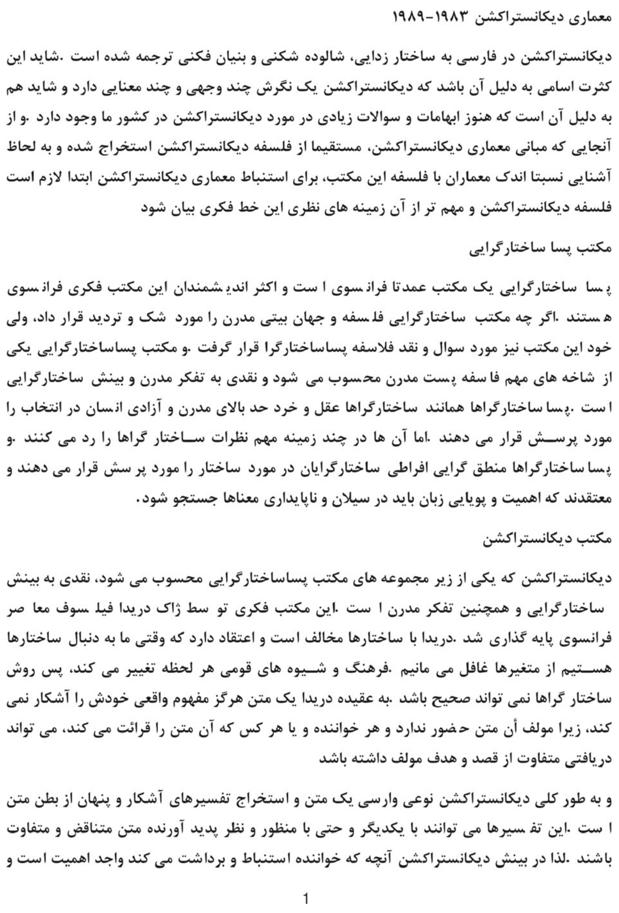
خلاصه اظهارنامه و اساسنامه شرکت مهوند تبریز
(سهامی‌خاص) که در تاریخ ۷۶/۱۲/۱۶ تحت شماره **۹۲۴۴** در
این اداره بثبت رسیده و در تاریخ ۷۶/۱۲/۱۲ از لحاظ امضاء ذیل
ثبت تکمیل گردیده برای اطلاع عموم در روزنامه رسمی و روزنامه
کثیرالانتشار آگهی می‌شود: ۱- موضوع شرکت: انجام کلیه کارهای
راه و ساختمان و زیربنایی شهری (سیویل) مطالعه- طرح و تهیه
نقشه‌های اجرایی راههای روستایی و شهرکها (نقشه‌برداری) طرح و
محاسبه و اجرای ابنیه فنی راه- انجام کلیه کارهای تأسیساتی اعم
از آبرسانی و زهکشی و سدسازی و توزیع و انتقال نیرو- طرحهای
جمع‌آوری و تصفیه فاضلاب- طراحی- سازه‌های فولادی و بتنی
خدمات بازرگانی ۲- مدت شرکت: از تاریخ ثبت بمدت نامحدود
۳-مرکز اصلی شرکت: تبریز چهارراه حکیم نظامی خیابان امامیه
کوچه حسینیه پلاک ۲۷/۱ کدپستی ۵۱۶۴۷،۴ - سرمایه شرکت:
مبلغ یک‌میلیون ریال منقسم به یکصد سهم ده‌هزارریالی بانام که
کلًا بحساب جاری ۱۰۲۲ واریز گردیده است ۵- اولین مدیران و
دارندگان حق امضاء: آقایان بهروز سلاطین‌سقین‌سرا بعنوان
مدیرعامل و عضو هیئت‌مدیره و غلامرضا عظیمی‌سقین‌سرا بسمت
رئیس هیئت‌مدیره و حسن سلاطین‌سقین‌سرا بسمت نایب رئیس
هیئت‌مدیره برای مدت یکسال انتخاب و کلیه قراردادها و اسناد
تعهدآور شرکت از قبیل چک و سفته و برات و غیره با امضاء
مدیرعامل و رئیس هیئت‌مدیره و در غیاب او نایب رئیس هیئت‌مدیره
و ممهور به مهر شرکت معتبر خواهد بود ۶- اختیارات مدیرعامل:
طبق اساسنامه ۷- بازرس اصلی و علی‌البدل: آقایان حسن
کوشک‌سرای‌اصل بازرس اصلی و محرم قلی‌پور بازرس علی‌البدل.
ش۴۰۴۹۵ مسئول ثبت شرکتهای ثبت تبریز

FIGURE 12 Sample of OIN dataset.



منتخبنا الوطني للرماي يطير ل معسكره في مصر في كس سيا لشباب اليد منتخبنا يهدر فرص الفوز عل
اليابان ويواجه اليوم الكويتي قرا ف تصفيات اسيا لمونديال في التصفيات الوروبي لمونديال بعد الخسار من
ايران شاب ردني يصاب بحال هستيري مدرب السويد لا يزال يتطلع للتهل رغم الخسار للمشارك في البطول
العربي بالجزار منتخبنا الوطني للرماي يطير ل معسكره في مصر غادر البلاد صباح مس الفريق الوطني للرماي
متوجها ل جمهوري مصر العربي لاقام معسكر تدريبي خلال الفتر من ال سبتمبر الجاري بالقاهر بعدها يتوجه
الفريق ال الجزار للاشتراك في البطول العربي للرماي والتي ستقام هناك خلال الفتر من سبتمبر وحت اكتوبر
القادم

FIGURE 13 Sample of Arabic OCR dataset.



معماری دیکانستراکشن ۱۹۸۳-۱۹۸۹

دیکانستراکشن در فارسی به ساختار زدایی، شالوده شکنی و بنیان فکنی ترجمه شده است .شاید این
کثرت اسامی به دلیل آن باشد که دیکانستراکشن یک نگرش چند وجهی و چند معنایی دارد و شاید هم
به دلیل آن است که هنوز ابهامات و سوالات زیادی در مورد دیکانستراکشن در کشور ما وجود دارد .و از
آنجایی که مبانی معماری دیکانستراکشن، مستقیما از فلسفه دیکانستراکشن استخراج شده و به لحاظ
آشنایی نسبتا اندک معماران با فلسفه این مکتب، برای استنباط معماری دیکانستراکشن ابتدا لازم است
فلسفه دیکانستراکشن و مهم تر از آن زمینه های نظری این خط فکری بیان شود

مکتب پسا ساختارگرایی

پسا ساختارگرایی یک مکتب عمدتا فرانسوی ا ست و اکثر اندیشمندان این مکتب فکری فرانسوی
ه ستند .اگر چه مکتب ساختارگرایی فلسفه و جهان بیتی مدرن را مورد شک و تردید قرار داد، ولی
خود این مکتب نیز مورد سوال و نقد فلاسفه پساساختارگرا قرار گرفت .و مکتب پساساختارگرایی یکی
از شاخه های مهم فا سفه پست مدرن محسوب می شود و نقدی به تفکر مدرن و بینش ساختارگرایی
ا ست .پسا ساختارگراها همانند ساختارگراها عقل و خرد حد بالای مدرن و آزادی انسان در انتخاب را
مورد پرســش قرار می دهند .اما آن ها در چند زمینه مهم نظرات ســاختار گراها را رد می کنند .و
پسا ساختارگراها منطق گرایی افراطی ساختارگرایان در مورد ساختار را مورد پر سش قرار می دهند و
معتقدند که اهمیت و پویایی زبان باید در سیلان و ناپایداری معناها جستجو شود.

مکتب دیکانستراکشن

دیکانستراکشن که یکی از زیر مجموعه های مکتب پساساختارگرایی محسوب می شود، نقدی به بینش
ساختارگرایی و همچنین تفکر مدرن ا ست .این مکتب فکری تو سط ژاک دریدا فیل سوف معا صر
فرانسوی پایه گذاری شد .دریدا با ساختارها مخالف است و اعتقاد دارد که وقتی ما به دنبال ساختارها
هســتیم از متغیرها غافل می مانیم .فرهنگ و شــیوه های قومی هر لحظه تغییر می کند، پس روش
ساختار گراها نمی تواند صحیح باشد .به عقیده دریدا یک متن هرگز مفهوم واقعی خودش را آشکار نمی
کند، زیرا مولف أن متن حضور ندارد و هر خواننده و یا هر کس که آن متن را قرائت می کند، می تواند
دریافتی متفاوت از قصد و هدف مولف داشته باشد

و به طور کلی دیکانستراکشن نوعی وارسی یک متن و استخراج تفسیرهای آشکار و پنهان از بطن متن
ا ست .این تف سیرها می توانند با یکدیگر و حتی با منظور و نظر پدید آورنده متن متناقض و متفاوت
باشند .لذا در بینش دیکانستراکشن آنچه که خواننده استنباط و برداشت می کند واجد اهمیت است و

1

FIGURE 14 Sample of synthetic dataset.

Microsoft Word files containing text with various fonts, styles, and sizes, and each page of the Word file was saved as a PNG image. An example of the synthetic dataset's images, with each page having dimensions of 1653 × 2339, is depicted in Figure 14. The synthetic dataset can be accessed at Reference 42.

Table 3 summarizes the datasets used in the TLD step, providing essential details. Notably, the reported details for OIN dataset in Table 3, correspond to text images only (i.e., after removing non-text images) (Table 4). In order to thoroughly assess the performance of our proposed method within the OCR system, we categorized the connected components (CCs) into two distinct types: dot-diacritic and the main text, as illustrated in Figure 15. The results of this categorization are presented in Table 9, showcasing the outcomes of our method's performance.

In the realm of Text Layout Detection (TLD), we meticulously followed established norms for dataset partitioning, dedicating 70% for training, 10% for validation, and 20% for testing. It's worth emphasizing that our TLD method, in contrast to learning-based approaches, operates independently of extensive datasets due to its non-learning nature. Instead, we leverage a more streamlined dataset for trial and error, seamlessly incorporated within the validation set. This strategic choice not only streamlines resource utilization but also distinguishes our method from deep learning counterparts.

TABLE 3 Dataset information.

	#of images	#of lines	#of big CCs (main text)	#of small CCs (dot or diacritic)	Total number of CCs
Official Iranian newspapers dataset	1920	32,642	1,292,544	1,057,536	2,350,080
Arabic dataset	50	1198	47,006	40,395	87,401
Synthetic dataset	962	24,195	1,158,531	873,570	2,032,101

TABLE 4 The accuracy of the proposed method in detecting textual and non-textual regions on OIN dataset.

Method	$\text{Acc}_{\text{text-region}}$	$\text{Acc}_{\text{nontext-region}}$
Layout Parser	78.31%	94.53%
SSD	87.51%	93.39%
YOLOv3	93.33%	94.02%
Faster R-CNN	93.67%	94.36%
Proposed method in DLA step	**94.77%**	**98.04%**

The bolded values represent instances where the accuracy is notably higher compared to other values.

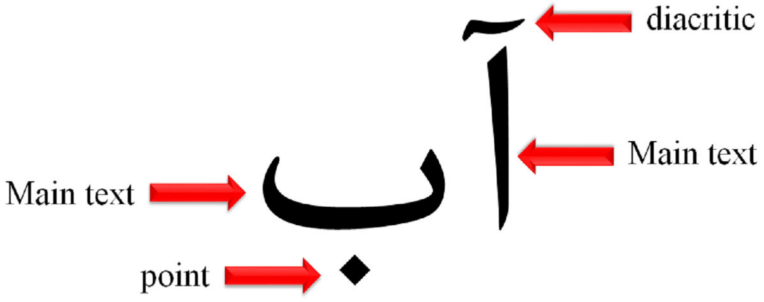


FIGURE 15 Difference between main text and dot or diacritic.

Furthermore, the decision to adhere to the standard dataset split in TLD becomes particularly relevant in the comparative analysis with deep learning-based methods. Unlike our TLD approach, these methods hinge on extensive training datasets for optimum performance. This nuanced detail should be underscored in the manuscript to elucidate the rationale behind our dataset split strategy. Conversely, in the realm of DLA, we maintained a consistent dataset configuration throughout training, ensuring methodological coherence. This deliberate decision, while seemingly straightforward, speaks to the robustness and reliability of our evaluation approach for the DLA stage.

4.2 | Comparative performance

4.2.1 | DLA step

In order to assess the performance of the proposed method in the DLA step, we conducted experiments on the OIN dataset. We compared the results with four other models based on deep learning: Faster R-CNN, YOLOv3, SSD, and Layout Parser. To evaluate the accuracy of each method, we employed Equations (3) and (4), which calculates the ratio of correctly recognized as text or nontext pixels. The obtained results are summarized in Table 1, where it can be observed that the proposed method achieved the highest accuracy among all the models.

$$\mathrm{Acc}_{\text{text-region}} = \frac{\sum_{j=1}^{\#\text{of pages}} \#\text{of pixels correctly detect as text region in page}_j}{\sum_{j=1}^{\#\text{of pages}} \#\text{of text region pixels in page}_j}, \tag{3}$$

$$\mathrm{Acc}_{\text{nontext-region}} = \frac{\sum_{j=1}^{\#\text{of pages}} \#\text{of pixels correctly detect as nontext region in page}_j}{\sum_{j=1}^{\#\text{of pages}} \#\text{of nontext region pixels in page}_j}. \tag{4}$$

To conduct thoroughly evaluate the proposed method in the DLA step, the PRImA dataset was utilized. The evaluation results of the proposed method and other methods on the PRImA dataset are presented in Table 5. Notably, the accuracy of the proposed method in both textual and non-textual region detection surpasses that of the other methods, highlighting its exceptional capability in accurately extracting both textual and non-textual regions. This remarkable performance can be attributed to utilizing a combination of methods, each excelling in extracting specific types of regions, with some specialized in extracting textual areas and others in extracting non-textual areas. The synergistic integration of these methods contributes to the overall effectiveness of the proposed method.

By achieving superior accuracy in the DLA step, the proposed method demonstrates its effectiveness in accurately identifying and separating text regions, providing a solid foundation for subsequent line extraction and overall OCR performance.

4.2.2 | TLD step

As already mentioned, we used three different datasets to compare the proposed method with state-of-the-art methods in the TLD problem. As shown in Table 6, we compared the performance of the proposed method with that of Kraken and OCRopus, which are based on deep neural networks, on the OIN dataset. As shown in Table 6, the proposed method achieved the best performance. These two compared methods had the best results in line extraction, so we have compared them with the proposed method. Also, the development team has updated these two methods many times, and they are among the best methods in the field of line extraction.

TABLE 5 The accuracy of the proposed method in detecting textual and non-textual regions on PRImA dataset.

Method	$\mathrm{Acc}_{\text{text-region}}$	$\mathrm{Acc}_{\text{nontext-region}}$
Layout Parser	85.47%	90.74%
SSD	89.32%	92.93%
YOLOv3	93.78%	95.73%
Faster R-CNN	93.53%	95.23%
Proposed method in DLA step	**96.11%**	**97.96%**

The bolded values represent instances where the accuracy is notably higher compared to other values.

TABLE 6 Text lines recognition error rate of the proposed method and methods based on deep learning on OIN dataset.

Method	The average error rate in detecting CCs per image	The average error rate in the incorrect removal of CCs per image	Total error rate	Average runtime per image (s)
OCRopus	0.66%	8.6%	9.26%	19.25
Kraken	2.14%	**0.17%**	2.32%	16.25
Proposed method	**0.19%**	0.31%	**0.51%**	**12.25**

The bolded values indicate lower error rates.

To evaluate each method, three evaluation criteria based on connected components have been used. The lines extracted by the proposed method are given to the input of the OCR system. In a line, the OCR system may encounter some errors by removing any CC, so we have defined criteria that include the final results of the OCR system. As shown in Equation (5), the first criterion indicates the error rate of each method in detecting the connected components in the line. In other words, if the connected components are placed incorrectly in the upper or lower line, they are considered as an error and are identified by this criterion. As shown in Equation (6), the second criterion calculates the error rate of each method in the incorrect removal of connected components, which are incorrectly considered background or removed as noise. Finally, as shown in Equation (7), the third criterion calculates total error rate of each method. In the third criterion, both of the previous criteria are involved. In addition to these three evaluation criteria, we have also considered each method's average runtime per image.

$$\text{First criterion}_i = \frac{\text{Number of CCs detected in wrong line}}{\text{Total number of CCs}} \times 100,\ \ i \in \text{methods}, \tag{5}$$

$$\text{Second criterion}_i = \frac{\text{Number of undetected CCs}}{\text{Total number of CCs}} \times 100,\ \ i \in \text{methods}, \tag{6}$$

$$\text{Third criterion}_i = \text{First criterion}_i + \text{Second criterion}_i,\ \ i \in \text{methods}. \tag{7}$$

The OCRopus method has drawbacks, including high running time and difficulty removing small objects and dots in the image. Figure 16A shows a line output from the proposed method, while Figure 16B shows the same line output using OCRopus, which removed many dots from the image. Persian and Arabic have more dots and diacritics than English.

Similar to OCRopus, Kraken has a high execution time. Furthermore, Kraken has low accuracy in skewed or curved images. In oblique images where most lines overlap in Kraken, the performance of the OCR system is significantly reduced. Figure 17 shows an example of the output of the Kraken method in dataset 3. As shown in Figure 17, most lines overlap and reduce the performance of the OCR system. Kraken and OCRopus have a significant problem detecting skewed or curved images. In contrast, the lines' skewness or curvature has a more negligible effect on our method. Our method is resistant to curvature and skew due to the use of the optimum font size.

Table 7 shows the error rates of methods in detecting and extracting lines on the synthetic dataset. As shown in Table 2, the proposed method has the lowest error rate, which indicates the excellent performance of the proposed method. In contrast, the error rate of OCRopus and Kraken methods on the synthetic dataset has increased compared to the OIN dataset. The higher error rate of these two methods on the synthetic dataset was the presence of images with the IranNastaliq font. As mentioned in Section 4.1, the synthetic dataset consists of six different fonts, which one of these fonts is IranNastaliq. Extracting lines in images that contain this font is more difficult due to its high resemblance to handwritten text. The proposed method solves the challenge by defining the concept of font size and extracting the lines correctly.

Figure 18 shows an example of the challenge facing OCRopus and Kraken methods in extracting lines from images with IranNastaliq font. As shown in Figure 18, both OCRopus and Kraken methods failed to extract the line correctly while the proposed method correctly extracts the line.

Due to the high complexity of the OIN dataset compared to the others, as shown Table 6, the error rates of the proposed method in this dataset is higher. In the OIN dataset, the lines are very close to each other and are overlapped in some cases. Skewed images and curved lines are other problems in this dataset which causes text-line extraction complex.

To further evaluate the performance of the proposed method, we tested the proposed method on the Arabic OCR dataset. As shown in Table 8, the proposed method had the lowest error rate. The OCRopus method still has the problem of incorrectly removing CCs in the Arabic OCR dataset. As shown in Figure 19, many dots are removed by this method, which is problematic in a language like Arabic, where words have many dots.

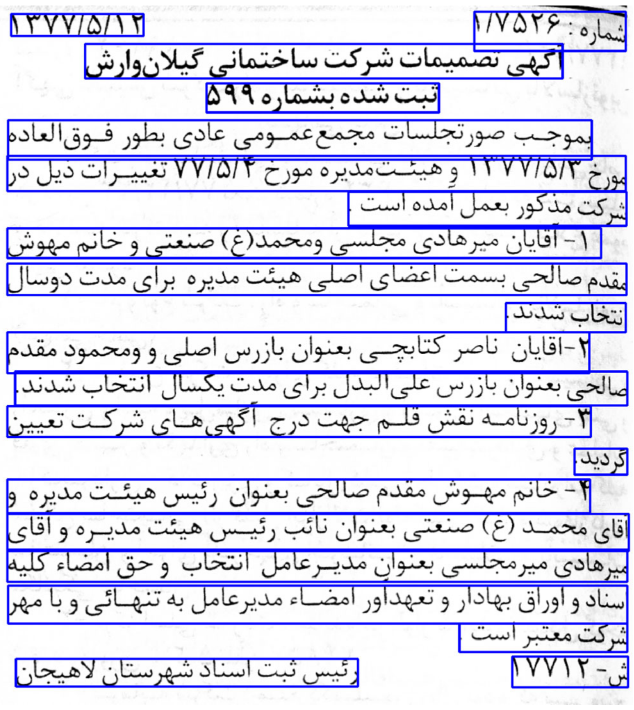
مورخ ۱۳۷۷/۵/۳ و هیئت‌مدیره مورخ ۷۷/۵/۴ تغییرات ذیل در

مورخ ۱۳۷۷/۵/۳ و هیئت‌مدیره مورخ ۷۷/۵/۴ نعبیرات دیل در

(A) output line in the proposed method

(B) Non curved line

FIGURE 16 Problem of removing small objects in OCRopus method for OIN dataset. (A) Output line in the proposed method. (B) Non curved line.



۱۳۷۷/۵/۱۲ شماره : ۱/۷۵۲۶

آگهی تصمیمات شرکت ساختمانی گیلان‌وارش

ثبت شده بشماره ۵۹۹

بموجب صورتجلسات مجمع‌عمـومی عادی بطور فـوق‌العاده
مورخ ۱۳۷۷/۵/۳ و هیئـت‌مدیره مورخ ۷۷/۵/۴ تغییـرات ذیل در
شرکت مذکور بعمل آمده است.
۱- آقایان میرهادی مجلسی ومحمد(ع) صنعتی و خانم مهوش
مقدم صالحی بسمت اعضای اصلی هیئت مدیره برای مدت دوسال
نتخاب شدند.
۲-اقایان ناصر کتابچـی بعنوان بازرس اصلی و ومحمود مقدم
صالحی بعنوان بازرس علی‌البدل برای مدت یکسال انتخاب شدند.
۳- روزنامـه نقش قلـم جهت درج آگهی‌هـای شرکـت تعیین
گردید.
۴- خانم مهـوش مقدم صالحی بعنوان رئیس هیئـت مدیره و
اقای محمـد (ع) صنعتی بعنوان نائب رئیـس هیئت مدیـره و آقای
میرهادی میرمجلسی بعنوان مدیـرعامل انتخاب و حق امضاء کلیه
سناد و اوراق بهادار و تعهداور امضـاء مدیرعامل به تنهـائی و با مهر
شرکت معتبر است.
ش- ۱۷۷۱۲ رئیس ثبت اسناد شهرستان لاهیجان

FIGURE 17 The output of the Kraken method on OIN dataset.

TABLE 7 Text lines recognition error rate of the proposed method and methods based on deep learning on synthetic dataset.

Method	The average error rate in detecting CCs per image	The average error rate in the incorrect removal of CCs per image	Total error rate	Average runtime per image (s)
OCRopus	2.35%	8.23%	10.39%	13.85
Kraken	5.3%	0.11%	5.32%	11.77
Proposed method	**0.01%**	**0.1%**	**0.11%**	**10.31**

The bolded values indicate lower error rates.

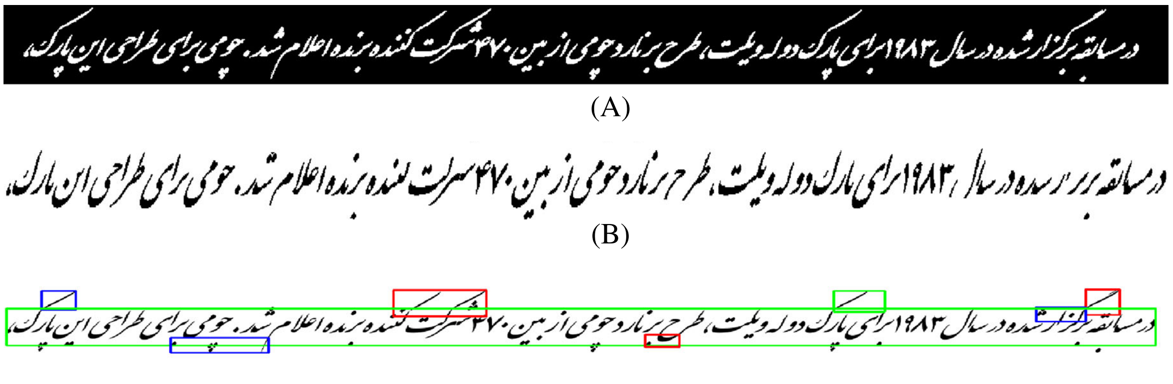
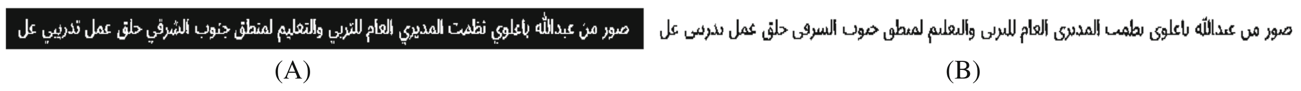

درمسابقه برگزار شده در سال ۱۹۸۳ برای پارک دوله ویلت، طرح برنارد چومی از بین ۴۷۰ شرکت کننده برنده اعلام شد. چومی برای طراحی این پارک،
(A)
(B)
(C)

FIGURE 18 Problem extracting lines in images with IranNastaliq font in synthetic dataset. (A) Output line in the proposed method. (B) Output line in the OCRopus method. (C) Output lines in the Kraken method.

TABLE 8 Text lines recognition error rate of the proposed method and methods based on deep learning on the Arabic OCR dataset.

Method	The average error rate in detecting CCs per image	The average error rate in the incorrect removal of CCs per image	Total error rate	Average runtime per image (s)
OCRopus	0.31%	3.82%	4.13%	3.23
Kraken	1.23%	**0.05%**	1.29%	3.66
Proposed method	**0.01%**	0.17%	**0.18%**	**3.11**

The bolded values indicate lower error rates.



صور من عبدالله باعلوي نظمت المديري العام للتربي والتعليم لمنطق جنوب الشرقي حلق عمل تدريبي عل

صور من عىدالله ىاعلوى ىطمى المدىرى العام للىرى والىعلىم لمىطق حىوى السرقى حلق عمل ىدرىى عل

FIGURE 19 Problem of removing small objects in OCRopus method for Arabic dataset. (A) Output line in the proposed method. (B) Output line in the OCRopus method.

TABLE 9 The effect of line detection step on the OCR.

	OIN dataset			Arabic dataset			Synthetic dataset		
	Dot or diacritic error	Main text error	Total error rate	Dot or diacritic error	Main text error	Total error rate	Dot or diacritic error	Main text error	Total error rate
Tesseract-OCR error rate without line detection step	8.71%	3.21%	6.235%	18.55%	7.37%	13.38%	7.54%	4.48%	6.22%
Tesseract-OCR error rate with line detection step	**5.24%**	**1.22%**	**3.431%**	**2.1%**	**0.85%**	**1.52%**	**4.34%**	**3.27%**	**3.88%**

The bolded values indicate lower error rates.

4.3 | The effect of line detection on the OCR

To prove that line detection has a significant effect on the accuracy of the OCR, first, we used Tesseract-OCR 5.1.0 without the line detection step to detect texts of the OIN and compared the results with when we used the line detection step. As shown in Table 9 using the line detection step before detecting characters with Tesseract-OCR 5.1.0 can recognize both types of CCs (main text and dot or diacritic) with higher accuracy.

5 | CONCLUSION

This study addressed the challenges associated with DLA and TLD in OCR systems, mainly focusing on the Persian language. We evaluated the performance of our proposed method by developing the OIN dataset, which contains challenging documents with curved lines, skewed images, and diacritics. We successfully extracted text and non-text regions with high accuracy by employing deep learning models in the DLA stage and implementing a voting system. Additionally, we introduced an angle correction method, an optimum font size concept, and an efficient algorithm to eliminate line curvature, significantly improving the accuracy of OCR systems. Comparative evaluations against state-of-the-art methods demonstrated the superiority of our approach in both DLA and TLD steps. Notably, our method achieved a 2.8% improvement in Tesseract-OCR 5.1.0 accuracy on the OIN dataset. The results of this study contribute to advancing OCR technology for Persian language documents and lay the foundation for further research in this domain. Future work should focus on expanding the dataset and exploring other languages with unique script characteristics to develop efficient methods for non-Latin languages. Overall, our findings highlight the significance of addressing DLA and TLD challenges to enhance OCR performance and pave the way for more accurate and reliable document analysis systems.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest regarding the publication of this article.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/eng2.12832.

DATA AVAILABILITY STATEMENT

The dataset generated during the current study are openly available in "PDN dataset" at https://drive.google.com/file/d/1mW42XHwY2hM4Z-ouhkEtgq8HJJq_IOib/view, Reference number 43.

ORCID

Amirreza Fateh https://orcid.org/0000-0001-9894-9131
Mansoor Fateh https://orcid.org/0000-0003-2133-3480
Vahid Abolghasemi https://orcid.org/0000-0002-2151-5180

REFERENCES

1. Li D, Hou J, Gao W. Instrument reading recognition by deep learning of capsules network model for digitalization in industrial internet of things. *Eng Rep*. 2022;4(12):e12547.
2. Poon C. Factors implicating the validity and interpretation of mechanobiology studies in simulated microgravity environments. *Eng Rep*. 2020;2(10):e12242.
3. Shawon MTR, Tanvir R, Alam MGR. Bengali handwritten digit recognition using CNN with explainable AI. *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE; 2022:1-6.
4. Zhang Z, Damiani E, Al Hamadi H, Yeun CY, Taher F. Explainable artificial intelligence to detect image spam using convolutional neural network. *2022 International Conference on Cyber Resilience (ICCR)*. IEEE; 2022:1-5.
5. Ehikioya SA, Zeng J. Mining web content usage patterns of electronic commerce transactions for enhanced customer services. *Eng Rep*. 2021;3(11):e12411.
6. Fateh A, Rezvani M, Tajary A, Fateh M. Providing a voting-based method for combining deep neural network outputs to layout analysis of printed documents. *J Mach Vis Image Process*. 2021;9:47-64.
7. Rahmati M, Fateh M, Rezvani M, Tajary A, Abolghasemi V. Printed Persian OCR system using deep learning. *IET Image Process*. 2020;14:3920-3931.
8. Fateh A, Fateh M, Abolghasemi V. Multilingual handwritten numeral recognition using a robust deep network joint with transfer learning. *Inform Sci*. 2021;581:479-494.
9. Guo Y, Sun Y, Bauer P, Allebach JP, Bouman CA. Text line detection based on cost optimized local text line direction estimation. *Color Imaging XX: Displaying, Processing, Hardcopy, and Applications*. Vol 9395. International Society for Optics and Photonics; 2015:1-7.
10. Bukhari SS, Shafait F, Breuel TM. Coupled snakelets for curled text-line segmentation from warped document images. *Int J Doc Anal Recognit*. 2013;16(1):33-53.
11. Amer IM, Hamdy S, Mostafa MG. Deep Arabic document layout analysis. *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE; 2017:224-231.
12. Ayesh M, Mohammad K, Qaroush A, Agaian S, Washha M. A robust line segmentation algorithm for Arabic printed text with diacritics. *Electron Imaging*. 2017;2017(13):42-47.
13. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2016:779-788.
14. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press; 2015:91-99.
15. Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M, eds. *European Conference on Computer Vision*. Springer; 2016:21-37.
16. Shen Z, Zhang R, Dell M, Lee BCG, Carlson J, Li W. LayoutParser: a unified toolkit for deep learning based document image analysis. arXiv preprint arXiv:2103.15348, 2021.
17. Redmon J, Farhadi A. Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
18. Antonacopoulos A, Bridson D, Papadopoulos C, Pletschacher S. A realistic dataset for performance evaluation of document layout analysis. *2009 10th International Conference on Document Analysis and Recognition*. IEEE; 2009:296-300.
19. Zhong X, Tang J, Yepes AJ. PubLayNet: largest dataset ever for document layout analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE; 2019:1015-1022.

20. Li M, Cui L, Huang S, Wei F, Zhou M, Li Z. TableBank: a benchmark dataset for table detection and recognition. arXiv preprint arXiv:1903.01949, 2019.
21. Lee BCG, Mears J, Jakeway E, et al. The newspaper navigator dataset: extracting headlines and visual content from 16 million historic newspaper pages in chronicling America. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM; 2020:3055-3062.
22. Shen Z, Zhang K, Dell M. A large dataset of historical Japanese documents with complex layouts. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE; 2020:548-549.
23. Mahmood A, Srivastava A. A novel segmentation technique for Urdu type-written text. *2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*. IEEE; 2018:1-5.
24. Ahmad I, Wang X, Li R, Ahmed M, Ullah R. Line and ligature segmentation of Urdu Nastaleeq text. *IEEE Access*. 2017;5: 10924-10940.
25. Soujanya P, Koppula VK, Gaddam K, Sruthi P. Comparative study of text line segmentation algorithms on low quality documents. *Int J Comput Sci Inf*. 2010;II:110-116.
26. Garg R, Garg NK. A new approach for line segmentation in Punjabi language using strip based projection profile method; 2014.
27. Nguyen TT, Dai Pham X, Kim D, Jeon JW. A test framework for the accuracy of line detection by Hough transforms. *2008 6th IEEE International Conference on Industrial Informatics*. IEEE; 2008:1528-1533.
28. Rais M, Goussies NA, Mejail M. Using adaptive run length smoothing algorithm for accurate text localization in images. In: San Martin C, Kim SW, eds. *Iberoamerican Congress on Pattern Recognition*. Springer; 2011:149-156.
29. Lyu B, Akama R, Tomiyama H, Meng L. The early Japanese books text line segmentation base on image processing and deep learning. *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE; 2019:299-304.
30. Fateh A, Fateh M, Abolghasemi V. Text line detection and correction for challenging datasets: a case study with newspapers dataset; 2023.
31. Latest release of Kraken; 2021. https://github.com/mittagessen/kraken
32. Latest release of OCRopus; 2017. https://github.com/ocropus/ocropy
33. Grüning T, Leifert G, Strauß T, Michael J, Labahn R. A two-stage method for text line detection in historical documents. *Int J Doc Anal Recognit*. 2019;22(3):285-302.
34. Koo HI. Text-line detection in camera-captured document images using the state estimation of connected components. *IEEE Trans Image Process*. 2016;25(11):5358-5368.
35. Malakar S, Halder S, Sarkar R, Das N, Basu S, Nasipuri M. Text line extraction from handwritten document pages using spiral run length smearing algorithm. *2012 International Conference on Communications, Devices and Intelligent Systems (CODIS)*. IEEE; 2012: 616-619.
36. Breuel TM. The OCRopus open source OCR system. *Document Recognition and Retrieval XV*. Vol 6815. International Society for Optics and Photonics; 2008:68150F.
37. Sabbour N, Shafait F. A segmentation-free approach to Arabic and Urdu OCR. *Document Recognition and Retrieval XX*. Vol 8658. SPIE; 2013:215-226.
38. Zhang K, Zuo W, Chen Y, Meng D, Zhang L. Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. *IEEE Trans Image Process*. 2017;26(7):3142-3155.
39. Bradley D, Roth G. Adaptive thresholding using the integral image. *J Graphics Tools*. 2007;12(2):13-21.
40. Youssef H. Arabic dataset OCR; 2020. https://drive.google.com/drive/folders/1-wsm4NIZB8Reu70jg-wBO56Pq89N6fs
41. Fateh A, Rezvani M, Tajary A, Fateh M. Persian printed text line detection based on font size. *Multimed Tools Appl*. 2022;82: 2393-2418.
42. Fateh A. Persian dataset in different font types, sizes, and styles; 2021. https://drive.google.com/file/d/1jaDp7qI6480yNImRZQpkYaOJ8o7mv8J/view?usp=sharing
43. Fateh A. Official Iranian Newspaper dataset; 2021. https://drive.google.com/file/d/1mW42XHwY2hM4Z-ouhkEtgq8HJJqIOib/view?usp=sharing

How to cite this article: Fateh A, Fateh M, Abolghasemi V. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. *Engineering Reports*. 2023;e12832. doi: 10.1002/eng2.12832