Scientia Iranica D (2020) 27(6), 3019–3033

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

A rule-based post-processing approach to improve Persian OCR performance

Z. Khosrobeigi[a], H. Veisi[a,*], H.R. Ahmadi[a], and H. Shabanian[b]

a. *Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran.*
b. *Department of Electrical and Computer Engineering, University of Memphis, Memphis, USA.*

Received 16 May 2019; received in revised form 20 June 2020; accepted 3 October 2020

KEYWORDS
Persian optical character recognition; Post-processing; Complex-structure images.

Abstract. Optical Character Recognition (OCR) is a system to convert images including text into an editable text. Nowadays, the accuracy of these systems in images with simple-structure and high quality is high. However, the performance degrades for images with complex-structure, low quality, and in the presence of noise, scratches, pictures, stamps, or other non-textual symbols. This paper proposes a Persian OCR post-processing technique to increase the accuracy of the OCR systems dealing with real-world challenging samples. The proposed method extracts five features in each line of the text and uses seven proposed rules to investigate whether that line should be ignored or not. To evaluate the proposed method, Khana (structural based) and Bina (deep learning-based) Persian OCR systems have been utilized, a dataset containing 200 complex-structure images has been collected, and a dataset including 100 simple-structure images has been used. The accuracy of Khana and Bina in images with a complex-structure is 39% and 58%, respectively, while after applying the proposed post-processing method, the accuracy increases to 93% and 91%, respectively.

1. Introduction

Optical Character Recognition (OCR) is a powerful technology to recognize and extract characters from image representation of documents, e.g. PDF files, images captured by a digital camera, and scanned paper documents; and convert it to a machine-readable text image [1]. This kind of system has several practical applications including data entry, process automation, aid to blind people, automatic license plate readers, automatic cartography, reading application forms [2], and scanned document retrieval [3]. Different types of OCR [4] involve different disciplines of computer science [5]. A general framework of an OCR system is presented in Figure 1.

According to the schematization of the whole system shown in Figure 1, the next step after data entry (digitally captured image of the text) is document analysis. Document analysis decomposes a given electronic image document into its component regions such as textual parts, tables, figures, and lines [7,8]. This process involves multiple procedures to gain an understanding of the logical structure of a document, e.g. identifying components of the moment, finding their relationships, and labeling their properties [9]. As a result of this step, the input image is divided into several segments which are important for the fur-

*. *Corresponding author.*
E-mail addresses: khosrobeygi.zohre@ut.ac.ir (Z. Khosrobeigi); h.veisi@ut.ac.ir (H. Veisi); hrahmadi@ut.ac.ir (H.R. Ahmadi); hshbnian@memphis.edu (H. Shabanian)

doi: 10.24200/sci.2020.53435.3267

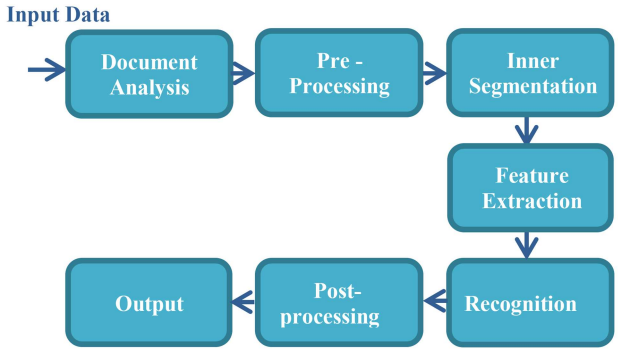


Figure 1. A general framework of an OCR system [6].

ther processes performed after the document analysis stage. In the pre-processing step, fully automatic text correction is achieved by noise attenuation and correction of image orientation. Afterwards, each word is broken up into its constituent characters/sub-words to satisfy the goal of internal segmentation. Then, the features of each character/sub-word are extracted. It is important to use features that are invariant to the character structure, font size, and font type. In the recognition phase, the extracted features from images of characters/sub-words are mapped to their equivalent units. Afterward, in the post-processing step, characters/sub-words are placed next to each other to create a context. Also, in this step, each constructed word is checked with a dictionary of words to correct possible misrecognized words [6,10].

The accuracy of OCR-generated text depends on various factors, e.g. the content of the text (complex or simple structure), the accuracy of each module of the system, and the resolution of the image. Moreover, situations, where images are not captured precisely (contained some deformations in the text), do not fit perfectly into the frame (some other objects are observable on the image) or the ink etc. is present on the paper, all play a destructive role on the precision of the algorithm. Additionally, the Persian language uses right to left scripts and consists of 32 letters. In order to create a word in the Persian language, characters should be connected, and based on the position that they joined (left, right, or both sides), their appearance might be changed. Also, some words include only one character. There are specific characteristics that have an impact on the OCR system performance and make this process more challenging than other languages such as English. Some of the most important challenges are as follows [11,12]:

- There is a character /G/ "گ" in the Persian language, which contains an oblique line at the top. This line is usually taken wrongly as a vertical line added to /K/ "ک", which is another character in this language;
- In the Persian language, several words consist of different numbers of dots. For example, /Zh/ "ژ", /P/ "پ", and /T/ "ت". This might lead to many errors in the character recognition process. For instance, the character /Zh/"ژ" can be detected as /Z/ "ز", which is a completely different character in the Persian language;
- There are some words like /seated/ "نشسته" which have many semicircles, called Dandaneh. Separation of each Dandaneh is difficult due to the semicircle detection step in the character recognition process;
- A character may have different shapes depending on where it appears in a word. For example, character /H/ "ه" is shown as /Guidance/ "هدایت" in the first letter of the word, as /Moonlight/ "مهتاب" in the middle of the word, and as /Blossom/ "شکوفه" at the end of the word, which also can be connected to other characters or not, such as /Simple/ "ساده" in which the character /H/ "ه" is not connected to its previous character [13];
- Each word in the Persian language consists of one or more sub-word(s), which are separated by a short space. Each sub-word includes one or more character(s). For instance, /Begin/ "شروع" has three sub-words and /Word/ "کلمه" has one sub-word. Hence, specifying each character by its vertical line is difficult for segmentation. Therefore, the segmentation of each word is done by sub-word(s), but the length of each sub-word is variable;
- Some documents include a combination of Persian words (which are right to left) and English words (which are left to right), which is also challenging in the segmentation process.

Among all the mentioned challenges, the most effective factor is the quality of the input image, which allows OCR to generate a text with more than 96% accuracy. On the contrary, by reducing the quality of the input image, the accuracy is decreased. A proper image resolution for the OCR systems is 300 DPI (dot per inch). However, sometimes the input data does not meet this requirement, e.g., when an image is captured by a smartphone with a low-quality camera.

Another issue is the recognition process in different languages. For example, in the Khana Persian OCR system, the OCR has the capability to detect characters in both English and Persian languages [14]. In this system, if a user selects the Persian language in the settings of the system and provides an image with English words, the output would be a sequence of meaningless letters and numbers. This is due to the absence of an intelligent automated pre-processing step (which can select an appropriate language). Similarly, it is possible that an image (without any text) is passed to the input of the algorithm, e.g. an image

of a forest. In this case, the OCR processes all color pixels and generates an equivalent text which consists of meaningless numbers and letters. This problem may also be solved with intelligent automated preprocessing.

The input images of an OCR system can be categorized into three classes, namely printed documents, handwritten documents, and printed-handwritten documents. The printed documents are documents that are first written by different word processing applications and then printed by a printer; the handwritten documents, on the other hand, are written by hand, and the printed-handwritten documents include both. For instance, in a registration paper form, some of the sections are printed (such as titles) and others are filled or signed by hand. This category of the input image is not supported by the Persian OCRs which are used in this paper and if tested, the results will be meaningless characters and numbers. The input image of the Persian OCRs used in this paper is limited to printed documents; therefore, there is a demand for having a Persian OCR with the capability of recognizing handwritten documents as an input image [15].

The existence of different elements in an input image such as tables, pictures, graphics [16], manuscripts, stamps, and other elements in which there is no text, can be categorized as the other challenges in the recognition process. We call these types of images "images with complex structure". For example, the scanned image of a student's transcripts, which contains a signature and a stamp is a good example of an image with a complex structure. Figure 2 demonstrates a complex structure image and Figure 3 shows the results of the two different Persian OCR systems, Khana [14] and Bina [17], without applying post-processing.

As shown in Figure 3, most of the lines have numbers and letters, which do not carry any meaningful information. These meaningless outputs are the results of recognizing the non-textual elements in the image which are not handled accurately by the segmentation module. Hence, there is a demand to improve these kinds of outputs by removing these meaningless numbers and letters.

Figure 4 depicts an example of the results after applying the proposed post-processing method. As obvious in the figure, most of the irrelevant lines have been removed from the results.

In addition, an image might include a graphical font, an example of which is shown in Figure 5. Similar to the previously mentioned problem, the OCR system would not show any meaningful result in the output of this image.

The above-mentioned challenges of the OCR output in the Persian text can be substantially improved by the recently proposed techniques on the segmentation, recognition, or post-processing phases. In this paper, we focus on post-processing results of the Persian OCR output and employ a rule-based approach to improve the performance of the results. It should also be noted that most of the issues mentioned above cannot be solved using a simple post-processing technique, and there is a demand to apply an intelligent and complicated post-processing approach.


شکل ۱۱ـ۲ـ تغییر کاربری اراضی کشاورزی

۲ـ فشار بر منابع آب: در سال‌های اخیر، با افزایش جمعیت و توسعهٔ صنایع، نیاز به آب افزایش پیدا کرده و فشار بر منابع آب سطحی و زیرزمینی شدت یافته است. با توسعهٔ فرهنگ صرفه‌جویی و استفاده از شیرآلات خودکار، مصارف بهداشت خانگی نیز به صورت بهینه درآمده است. در بخش صنعت نیز با افزایش تعرفه و تشویق آنان به استفاده از آب بازیافتی می‌توان فشار بر منابع آب را کاهش داد.

شکل ۱۲ـ۲ـ کاهش ذخیرهٔ آب در پشت سد

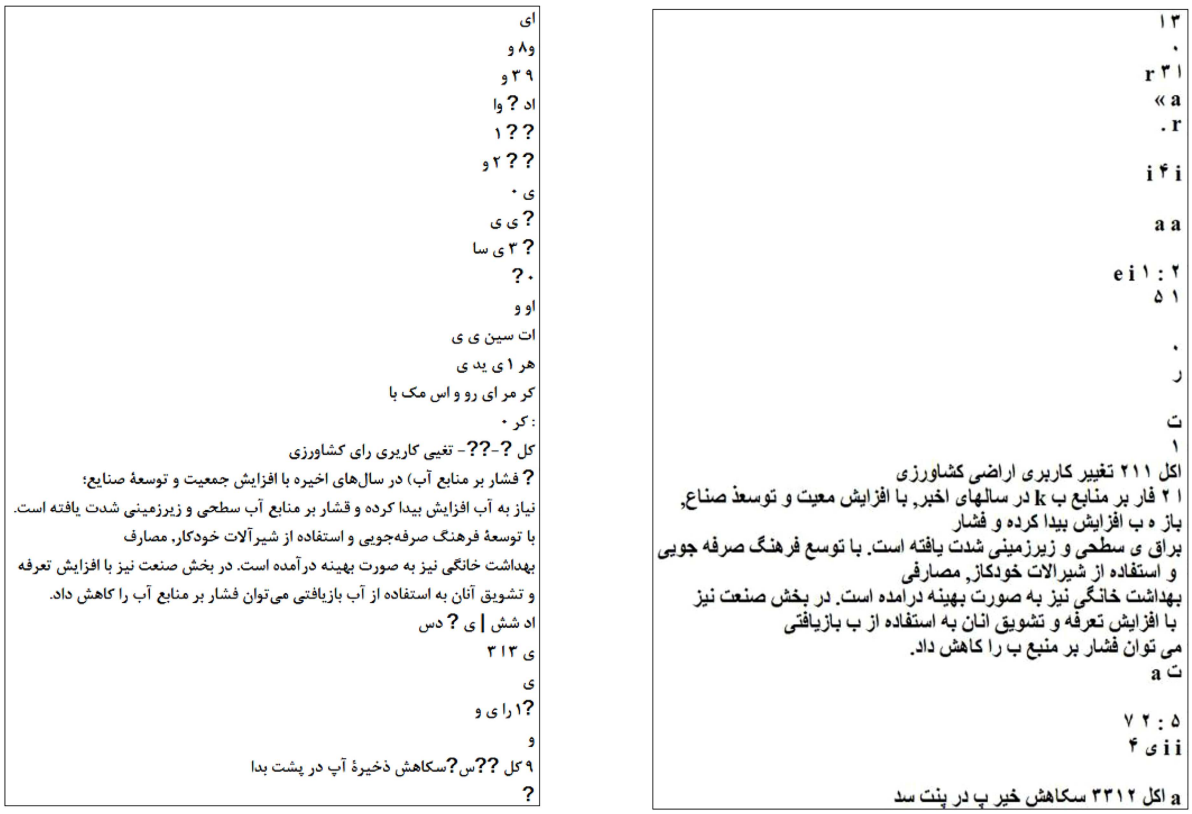
۳۸

Figure 2. An example of an image with a complex structure.

The rest of this paper is organized as follows: Section 2 describes recent concepts and available methods. Section 3 explains the details of the proposed method, while experimental results and evaluation metrics are illustrated in Section 4. Finally, discussions and conclusions are given in Section 5.

2. Related works

Several researchers investigated OCR post-processing methods to detect and correct various kinds of errors in the produced text document. Errors may occur in tokenization, boundary detection, character recognition, or other stages of the OCR. These methods can be categorized into three groups of different approaches, namely manual error correction, dictionary-based correction, and context-based correction approaches.



(a) (b)

Figure 3. The results of applying Khana and Bina OCR systems on the complex- structure image shown in Figure 2: (a) Results of the Khana OCR system and (b) results of the Bina OCR system.

Figure 4. Results of applying the proposed post-processing method to the Persian OCR results shown in Figure 3.

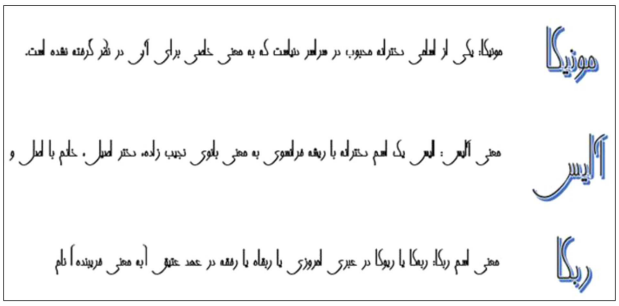


Figure 5. An example of a graphical font in a Persian text document.

2.1. Manual error-based post-processing approaches

This category of techniques solves a common error in the OCR recognition process, which is spelling mistakes in the output text. For example, it is possible to observe the number "6" instead of character "G" as an incorrect result of OCR.

The simplest technique to remedy this problem, which is obviously costly and time-consuming, requires a group of people to manually check and correct the OCR output. Although this technique can solve real world and non-word errors, due to the fact that optical illusions may cause some mistakes in the human eye, this technique is not considered a convincing way to correct errors in the OCR output text in large volumes of data.

2.2. Dictionary-based error post-processing approaches

These techniques automatically detect and correct OCR output errors using a lexicon [18,19]. This traditional strategy for spelling correction has been studied for several decades and is known as isolated

word error correction [20]. While this approach can handle the actual problem by correcting the OCR recognized words (when they are not present in the dictionary), it does not consider the context in which the error has occurred. Therefore, there is a demand to assemble a wide-ranging dictionary to make these approaches the acceptable solution for OCR results, which can even be applied to historical documents including names of geographical locations and specific terminologies.

2.3. Context-based error post-processing approaches

To overcome some of the limitations of dictionary-based techniques, the context of text has received more attention. Accordingly, context-based approaches are considered as an integrated version of the dictionary-based approaches [21–23]. In this category of the post-processing approaches, grammatical and semantic context errors are detected. A good example of these approaches, presented by Bassil et al. [24], used the Google online spelling suggestion to get common spelling suggestions on the English and Arabic languages. In this integrated version of the dictionary-based approach, the OCR error correction was performed by the "did you mean" spelling suggestion feature of the Google online web search engine. First, they performed word tokenization, then the Google search engine was used to receive a search query (consisting of the created tokens) to provide possible corrections.

As another example, some English OCRs first employ a pre-processing step to detect tables and images of the input document. These OCR algorithms then temporarily ignore the detected tables and images. In the end, the output of the OCR is filled by the corresponding tables and images, from the input document [25,26]. Contrary to English OCR systems, most Persian OCR systems do not include pre-processing steps [15] and require substantial improvements and refinements in their generated results.

The study of post-processing on the OCR results in the two latter categories (dictionary-based and context-based categories) has been undertaken for several different languages (Nagata in Japanese [27], Afli et al. in French [28], Kesorn et al. in Thai [29], Abu Douch et al. [30] and Magdy et al. [31] in Arabic, Ramanan et al. in Tamil [32], Kolak et al. [33] in Spanish, Igbo, Cebuano, and Arabic). As an example of the OCR post-processing in the Arabic language, Zaiz et al. [34] improved the Arabic OCR results by proposing a post-processing technique that worked based on a Support Vector Machine (SVM) classifier and a Puzzle algorithm. In that algorithm, a Levenshtein distance was calculated to detect the closest words. According to the obtained distance, the system decides to consider a match or employ a puzzle algorithm to perform some processes leading to the recognition of the matching word.

Among all currently widely spoken languages, only the Arabic language has some similarities with the Persian language [35–37]. For instance, there are 28 similar letters in both languages and the Persian language uses some modified variants of Arabic letters; character /k/ "ک" in Arabic is written "ک" in Persian. Despite the existing similarities, the grammar and the number of letters in the alphabets are not the same (there are four extra letters in the Persian language that do not exist in the Arabic language) [38]. These differences between the two languages make a substantial discrepancy in the methodology of the presented post-processing algorithms. Hence, the proposed post-processing methods in the Arabic language do not provide reasonable accuracy in the Persian OCR output, and so there is a demand to develop a new post-processing algorithm for the Persian OCR output.

In the literature, only a few attempts have been made for the Persian language. Accordingly, some of the works that have been done in Persian OCR will be described, and then the approaches using a post-processing contribution will be the focus in the results of the Persian OCR.

The first attempt at Persian OCR was in 1981 by Parhami and Taraghi [39] who proposed a Persian OCR system for a large font size text. A few years later, a new segmentation algorithm was presented by Azmi and Kabir [40] applying to Persian text in 20 different fonts. Based on their demonstrated results, the algorithm could achieve an accuracy of 98.5% in terms of the connected characters, which were recognized correctly. In [5], Menhaj and Adab improved the Persian OCR segmentation and recognition process using multilayer feedforward neural networks with Fourier descriptors covering multi-size Persian/Arabic text. In another attempt, Ebrahimi and Kabir [41] exploited the k-means algorithm to cluster printed Persian sub-words (each sub-word was assumed as a shape). In [12], Khosravi and Kabir introduced an integrated Persian OCR system based on blackboard architecture, which worked on 10 different fonts. The presented results showed an average recognition rate of 97.05% and 99.03% on word level and character level, respectively.

To the best of the authors' knowledge, the only post-processing approach to Persian OCR is based on some heuristic functions undertaken by Azadnia [42]. In that work, the author proposed an expert system to process the Persian texts generated by an OCR system. The system utilized some heuristic functions to model and find incorrect words and correct the detected errors using a suitable lexicon, which incorporates some suggested words. Based on the reported results, the

automatic correcting text showed 52.9% accuracy in recognizing and correcting the errors in the text.

As discussed earlier, OCR systems use various methodologies to perform pre-processing, recognition, and post-processing. In most Persian OCR systems, such as OCRs that have been analyzed and/or presented by Ghanbari in [15], the pre-processing step does not include any item detection. On the other hand, most of the users prefer using open source and free OCR systems with mediocre accuracy for image documents only containing text. These popular and well-spread systems generate meaningless characters when they receive images with a complex structure as the input. Hence, applying a post-processing step is essential and beneficial in the majority of OCR systems. To evaluate the necessity of applying a post-processing step, we used Khana and Bina OCR systems. Although document analysis and pre-processing steps are considered in these systems [14], the results show that substantial improvements are needed to obtain accurate results for the complex-structure input images.

In this study, a new Persian OCR post-processing approach is proposed which removes the need for a very large dataset because it eliminates meaningless letters and the need to manipulate them. The algorithm utilizes Persian grammar and structure to detect meaningless words. For instance, in the Persian language, the number of words with only one letter is very low. Hence, if there are a large number of one-letter words in a single sentence, it means that it includes some meaningless words and needs to be removed. Another observation is that, based on the structure of Persian language sentences, there should be a small number of numerals in the text lines. Consequently, an increase in the number of numerals in a text line indicates the presence of meaningless words.

The proposed post-processing method is suitable for images with complex structures. The reason is that the OCR output of the images with complex structures contains a large number of meaningless words (because of the existence of noise, tables, stamps, etc.), which need to be eliminated rather than manipulated. The elimination of the meaningless numbers and words from the OCR output substantially improves the speed of the post-processing algorithm, and it is considered an important factor when dealing with a large amount of data.

3. The proposed method

In this paper, a context-based post-processing technique on the results of Persian OCRs is proposed. In this technique, as per the results of Khana [14] and Bina [17], Persian OCRs are passed as input to the proposed algorithm. After line by line processing of the text, the post-processing algorithm generates final results through detecting and removing meaningless lines.

The proposed technique extracts five features in each line of the text and applies seven rules to the extracted features for post-processing the OCR results. Lines with poor accuracy are selected and removed using the proposed rules. The details of the suggested technique will be discussed comprehensively in this section. At first, both word tokenization and feature extraction steps will be reviewed, then the proposed rules will be discussed.

3.1. Word tokenization

For tokenizing and preparing the documents for feature extraction, the following steps are performed.

1. Each line is divided into its continuant words using the space character;
2. This step evaluates whether each word exists in the dictionary or not. If a word is found in the dictionary, a counter denoting the number of words contained in the dictionary is increased, and then the number of words with one or two letters are counted. Also, the number of words with more than two letters are counted and the word is removed from the list of line vocabularies. Persian OCR systems do not create any semi-space, hence, all the words in the output are separated by a space. For this reason, the proposed technique performs the tokenization step using the space character. In addition, there are not any words with half-space or space in the dictionary. For example, the dictionary does not contain single words that include spaces, such as /does/ "می کند", and only contains single words without spaces, such as "کند". Therefore, separating words based on existing space is not affecting the word itself. Moreover, words without their prefix are searched for in the dictionary. A token would be considered as a word with a long length if the word is not in the dictionary. Therefore, it is considered as a correct line;
3. If a word is Out Of Vocabulary (OOV), it is analyzed whether it is considered a number or not. If the word is a number, it is counted as a number and it is removed from the list of line vocabularies;
4. After evaluating all words in the line, OOV words are separated from their non-letter characters. This is due to the fact that all unwanted characters such as short space، numerals، and punctuation are considered separate words. For example, the word /is./ "است." is split to "است" and a /dot/ ".", or in /Number3/ "شماره3" ,"شماره" is separated from "3". After this separation, step 2 is repeated for

the newly separated words. However, if it does not exist in either, that word is considered as OOV;

5. The steps are repeated up to the end of the line. After checking all the words in the line, features are extracted.

3.2. Feature extraction

As previously mentioned, five features are extracted from each line of the Persian OCR output. In this step, at first, the number of words in each line is calculated as follows:

$$LenLine = \#\ Total\ words\ in\ the\ line. \tag{1}$$

Then, the dictionary likelihood for each line is determined based on the ratio of the number of existing words (in the dictionary) to the entire words of the line, which can be described as:

$$P_{Dic} = \frac{|Word_{\text{In } line} \cap Word_{in\ Dic}|}{LenLine}, \tag{2}$$

where $Word_{\text{In } line}$ is the number of words in a line, $Word_{in\ Dic}$ denotes the words in the dictionary and $|.|$ defines the cardinality.

Afterwards, the likelihood of a numeral existing in each line with respect to an entire word in that line is calculated as follows:

$$P_{Digit} = \frac{|Digit_{\text{In } line}|}{LenLine}, \tag{3}$$

where $Digit_{\text{In } line}$ indicates the number of numerals in the line.

Then, based on the number of letters of each word, the probability of short word occurrence is calculated:

$$P_{Short} = \frac{|Word_{Len=2} \cup Word_{Len=1}|}{LenLine}, \tag{4}$$

where $Word_{Len=2}$ shows words consisting of two letters and $Word_{Len=1}$ indicates one-letter words in the line.

Subsequently, the probability of long word occurrence is computed using the number of words with more than two letters with respect to the entire number of words in a line, which can be expressed as:

$$P_{Long} = \frac{|Word_{Len>2}|}{LenLine} = 1 - P_{Short}, \tag{5}$$

where $Word_{Len>2}$ infers the number of words with more than two letters in the line.

3.3. Post-processing rules

After extracting the features, acceptance or rejection of each line of the text (generated by Persian OCR) is decided using the proposed rules. In the rules, it is required to define a threshold as the acceptance level for each feature or feature combination. All thresholds provided in this paper have been defined in an ad-hoc manner based on the experience of working with the Khana Persian OCR system [14]. The authors have been studying and working on countless images generated by the Khana system using various types of input images such as news pages, textbooks, registration forms, etc. At first, a prior value indicating a suitable range for each threshold was found after comprehensive analysis and testing. Then, the final value was obtained after a few tests in the range of the prior value. Hence, the obtained thresholds are appropriate and practical for images with a variety of structures. In some cases, it is possible that changing the value of the threshold could result in better performance for a specific kind of input image, however, it will not be suitable for all the varieties of the input images. Therefore, the selected values for all thresholds satisfy a wide range of input images. In the following rules, if the corresponding equation for any rule (Rules 2 to 7) is satisfied for each line, then that line will be removed.

Rule 1: If the dictionary likelihood for each line computed by Eq. (2), is greater than 0.5, the corresponding line is presented in the output as a valid result. The reason is, if half the words of a line exist in the dictionary, the output will be valid. This assumption substantially speeds up the post-processing;

Rule 2: If the probability of long word occurrence in Eq. (5) is less than a predefined threshold of 0.2, and the dictionary likelihood of Eq. (2) is less than 0.5, the corresponding line is removed and will not be represented in the output (Eq. (6)):

$$P_{Long} < 0 \cdot 2 \quad \text{and} \quad P_{Dic} < 0 \cdot 5. \tag{6}$$

Figure 6 demonstrates an example of lines that have been removed by applying Rule 2;

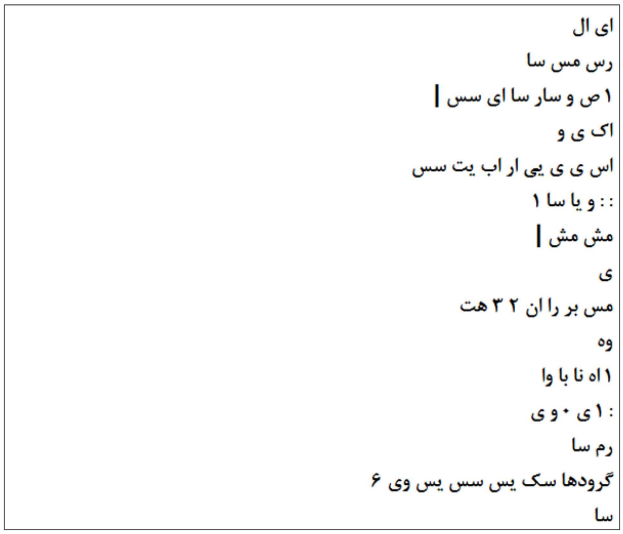


Figure 6. An example of the removed lines that were satisfied with the conditions of Rule 2.

Rule 3: If the value of the probability of being a numeral (Eq. (3)) is greater than a predefined threshold of 0.4, the value of dictionary likelihood (Eq. (2)) is less than a predefined threshold of 0.4, and the probability of long word occurrence (Eq. (5)) is less than a predefined threshold of 0.3, then the corresponding line is removed. The rule is summarized in Eq. (7):

$$P_{Digit} > 0 \cdot 4 \quad \& \quad P_{Long} < 0 \cdot 3 \quad \& \quad P_{Dic} < 0 \cdot 4. \quad (7)$$

Figure 7 shows an example of lines that are removed using Rule 3;

Rule 4: If the probability of long word occurrence (Eq. (5)) is less than the probability of short word occurrence (Eq. (4)), the probability of being a numeral (Eq. (3)) is less than the probability of short word occurrence (Eq. (4)), the value of dictionary likelihood (Eq. (2)) is less than a predefined threshold of 0.2, and the length of the line (Eq. (1)) is less than 9, then the corresponding line is removed. The rule is summarized in Eq. (8):

$$P_{Long} < P_{Short} \quad \text{and} \quad P_{Digit} < P_{Short},$$
$$P_{Dic} < 0 \cdot 2, \quad \text{and} \quad LenLine < 9. \quad (8)$$

This rule indicates that if the number of short words is more than the number of long words and numbers, and only a few words are in the dictionary, it specifies that the line does not carry meaningful information. It is because of the existence of a large number of short words in one line. Figure 8 shows an example of lines that are removed using Rule 4;

Rule 5: If the sum of the two probabilities of being a numeral (Eq. (3)) and short word occurrence (Eq. (4)) is greater than a predefined threshold of 0.5, the value of dictionary likelihood (Eq. (2)) is less than a predefined threshold of 0.1, and the length of the line (Eq. (1)) is less than 10, then the corresponding line is removed. This can be shown as Eq. (9):

$$(P_{Short} + P_{Digit}) > 0 \cdot 5, \quad \text{and} \quad P_{Dic} < 0 \cdot 1,$$
$$\text{and} \quad LenLine < 10. \quad (9)$$

The motivation of this rule is that when the sum of the number of words with one or two letters and numerals is greater than half the words in a line, and the words of the line are mostly OOV, it means that the corresponding sentence does not carry meaningful information. Hence, the line is removed.

An example of the lines that can be removed using Rule 5, is given in Figure 9.

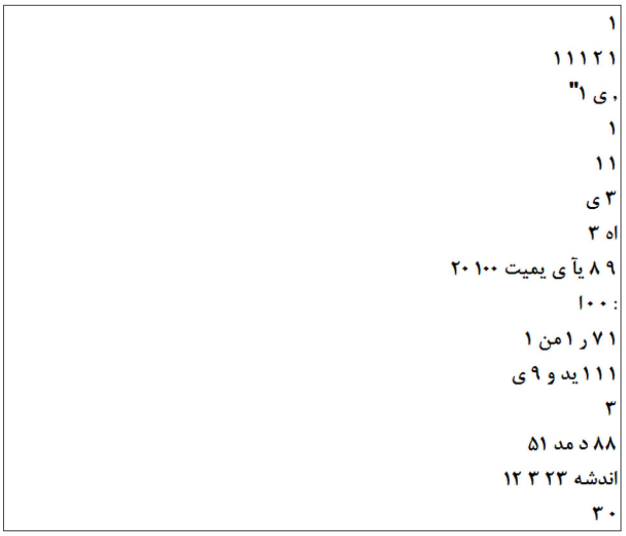


Figure 7. An example of the removed lines satisfying the conditions of Rule 3.

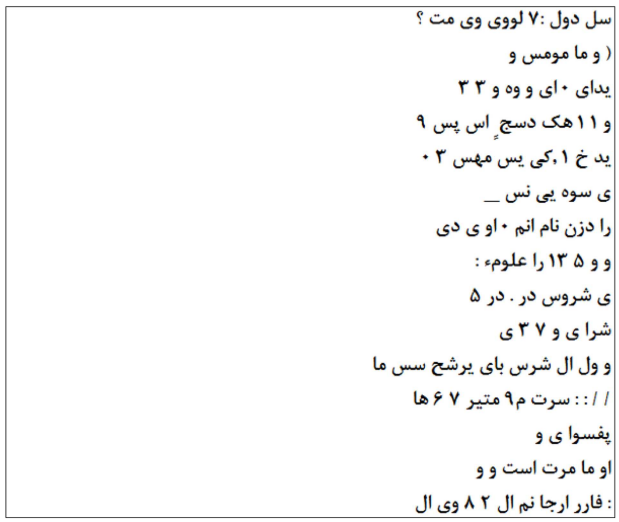


Figure 8. An example of the lines satisfying the conditions of Rule 4.

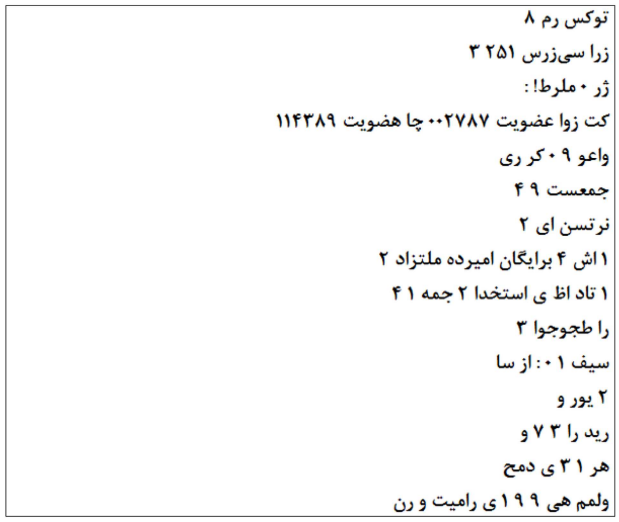


Figure 9. An example of the lines satisfying the conditions of Rule 5.

Rule 6: As summarized in Eq. (10), if the probability of long word occurrence (Eq. (5) is less than a predefined threshold of 0.3, the length of the line (Eq. (1)) is greater than 9, or the value of dictionary likelihood (Eq. (2)) is less than a predefined threshold of 0.2, and the length of the line (Eq. (1)) is greater than a predefined threshold of 9, then the corresponding line is removed:

$$(P_{Long} < 0 \cdot 3 \quad \text{and} \quad LenLine > 9),$$

or:

$$(P_{Dic} < 0 \cdot 2 \quad \text{and} \quad LenLine > 9). \tag{10}$$

The reason behind this rule is that in a long sentence we expect the constituent words to have a relatively high probability of being present in the dictionary. In sentences with more than 9 words, if 20% of the sentence is not in the dictionary, it is considered a meaningless sentence. Also, the first part of the rule dictates that in sentences with more than 9 words, if the probability of words with three or more letters is less than 30%, then the line must be deleted. The reason is that in long sentences, it is expected that in the worst-case scenario, there should be more than three (defined threshold) words with three or more letters. Figure 10 shows an example of this rule.

Rule 7: This rule checks three conditions. At first, the sum of the two probabilities of being a numeral (Eq. (3)) and short word occurrence (Eq. (4)) is calculated. If its difference from the probability of long word occurrence (Eq. (5)) is less than a predefined threshold of 2.5, the value of dictionary likelihood (Eq. (2)) is less than 0.12, and the length of the line is less than 10, then the corresponding line is removed. This rule can be summarized as Eq. (11):

$$(P_{Short} + P_{Digit}) - P_{Long} < 2 \cdot 5,$$

$$P_{Dic} < 0 \cdot 12, \quad \text{and} \quad LenLine < 10. \tag{11}$$

The Persian language contains numbers and words with two or more letters. In addition, the number of words with more than two letters is greater than the number of words with two letters and numbers. The inference of this rule is if the line has a single letter, double letter, and numbers more than a long word, it is considered a wrong line. An example of the results using Rule 7 is shown in Figure 11. If the extracted features of a line do not satisfy any of the mentioned rules (Rules 2 to 7), the corresponding line is presented in the output as a valid result.

Also, it is important to run the rules in the defined order. Figure 12 shows the pseudocode of applying the rules in an ordered manner. As shown in Figure 12, the first rule indicates that the line is considered as a valid

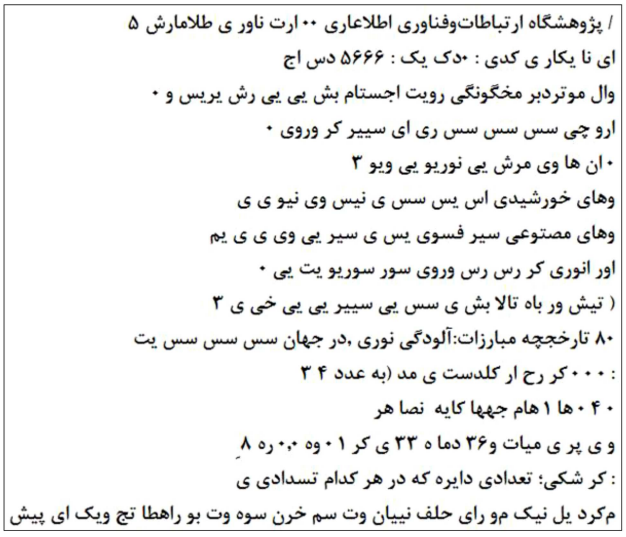


Figure 10. An example of the lines satisfying the conditions of Rule 6 and considered to be removed.

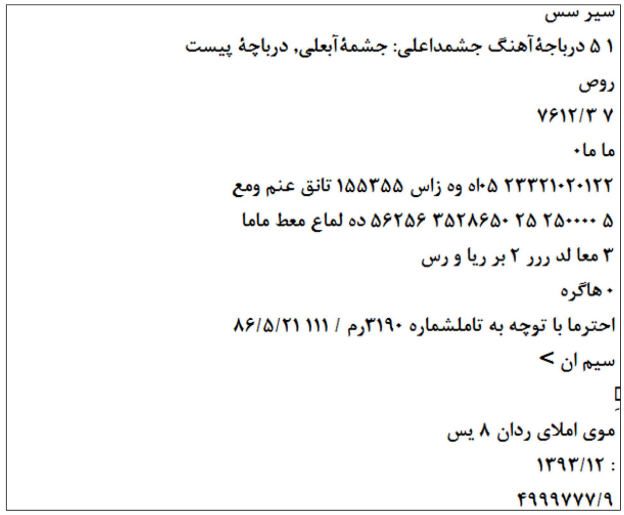


Figure 11. An example of the lines satisfying the conditions of Rule 7 and considered to be removed.

```
Input: Text_In (a text of an OCR)
Output: Text_Out (postprocessed input)
Text_Out = Ø
For each line in Text_In
    Calculate features using Eqs. (1) to (5)
    In P_Dic > 5
        Add line to Text_Out
    Else if P_Large < 0.2 & P_Small > 0.5 & P_Dic < 0.5
        Ignore line
    Else if P_Digit > 0.4 and P_Dic < 0.4 and P_Larg < 0.3
        Ignore line
    Else if P_Large < P_Small and P_Digit < P_Small and P_Dic < 0.2 and LenLine < 9
        Ignore line
    Else if (P_Small + P_Digit) > 0.5 and P_Dic < 0.1 and LenLine < 10
        Ignore line
    Else if (P_Large < 0.3 and LenLine > 9) or (P_Dic < 0.2 and LenLine > 9)
        Ignore line
    Else if (P_Small + P_Digit) − P_Large < 2.5 and P_Dic < 0.12 and LenLine < 10
        Ignore line
    Else
        Add line to Text_Out
```

Figure 12. Pseudo-code of the proposed post-processing algorithm.

output if more than half the words in the line exist in the dictionary. By using this first rule, the runtime of the algorithm decreases. Subsequently, if none of the Rules 2 to 7 are satisfied by a line of the text, it will also be considered as a valid output.

4. Evaluation and experimental results

4.1. Evaluation scope

In order to convert images into text, Bina [17] and Khana [14] Persian OCR systems (developed at the Digital Signal Processing (DSP) laboratory at the University of Tehran based on the Tesseract [43,44]) were utilized. The input image of the Khana Persian OCR system is an image in Persian or English languages. In this system, the image is converted to a binary format and after recognizing its structure, each line is converted to the corresponding text using a Long Short-Term Memory (LSTM) recursive neural network. Khana utilizes a font segmentation technique to divide the input image into two groups, namely: large and small fonts. In the group of large fonts, the pre-processing module removes small segments, which results in ignoring several parts of the output. Therefore, the generated output is a clean text. This pre-processing step also removes those words that are partially selected or written in small fonts, which shows more details in the output results of the OCR system. The reason is that the small-font configuration of the segmentation technique only removes a small number of existing numerals and words which represents more details compared to the large-font configuration. Hence, this case generates incomplete and meaningless words in the output, requiring post-processing to increase the readability and legibility of the text.

The Bina OCR uses a Convolutional Neural Network (CNN) to extract features of the input document images. Then, the extracted features are passed to an artificial recurrent neural network called bidirectional Long Short-Term Memory (LSTM) to perform the text recognition step. This system shows accuracies of greater than 90% and 85% for Persian only, and Persian-English documents, respectively [17]. Accuracies higher than 85% in Persian-English documents show the superiority of this OCR system with respect to other methods. This OCR system was trained for 11 fonts in three different sizes [17].

4.2. Datasets

4.2.1. Dictionary

In this study, two Persian OCR systems, named Khana [14] and Bina [17] have been utilized. Moreover, the authors of this paper utilized a tokenization technique to consider only one part of multi-unit token words. For instance, in the Persian language, there are many multi-unit token words, such as /Metavanad/"مـی تـوانـد" (equivalent to the verb "can" in English) that can be represented in three forms: "تواند می" or "می‌تواند" or "میتواند", in which there is a space, a half-space, and no space between "مـی" and "تواند", respectively. After tokenization, "می" is separated from "تواند" and the dictionary includes only the words "تواند" and "می". Use of this tokenization technique reduces the size of the dictionary significantly, due to removing the possible variations.

In addition, because of the existence of "date format" in the output text of the OCR, a tokenization process is applied based on all non-alphanumeric letters excluding "/", to avoid any impact on the "date" and to obtain the dates in the result as well. Otherwise, "date format" would be divided into three separate numbers, which would increase the probability in the rule condition "being a numeral" (Eq. (3)), and cause incorrect removal of the corresponding line.

4.2.2. Test dataset

In order to evaluate the proposed approach, a test dataset is created during the time of this research. It contains a subset of images that have been collected from various sources including:

- Submitted images by trial users to Khana Persian OCR online demo (The online demo of Khana Persian OCR system is available from its website, khanasoft.com);
- Samples from a scanned document analysis project from the Vice-President of Planning & Information Technology of the University of Tehran;
- Selected images from the Khana project dataset.

The test set samples are scanned or captured using cell-phone cameras by users, and therefore, includes various types and qualities. Some image samples are captured in imperfect lighting and contain skewness in borders, which challenge OCR performance. Due to the variations in the test samples, the samples were roughly categorized into two groups: complex-structure and simple-structure images [45]. An image with a simple structure only consists of text and does not have any non-textual data (an example is shown in Figure 13).

An image with a complex structure is shown in Figure 14, which includes a combination of picture, stamp, signature, and table, in addition to the text. Another example of a complex-structure image is demonstrated in Figure 15(a), along with the result of applying the Khana OCR system, which is shown in Figure 15(b). As seen in Figure 15(b), the generated result of applying the Khana OCR system on an image containing handwritten words, is mostly meaningless characters. The OCR systems in this research only convert the images of printed text and do not support handwritten text, and therefore, the handwritten sections of an image are ignored or considered to be noise.

Figure 13. An example of an image with a simple structure [45].

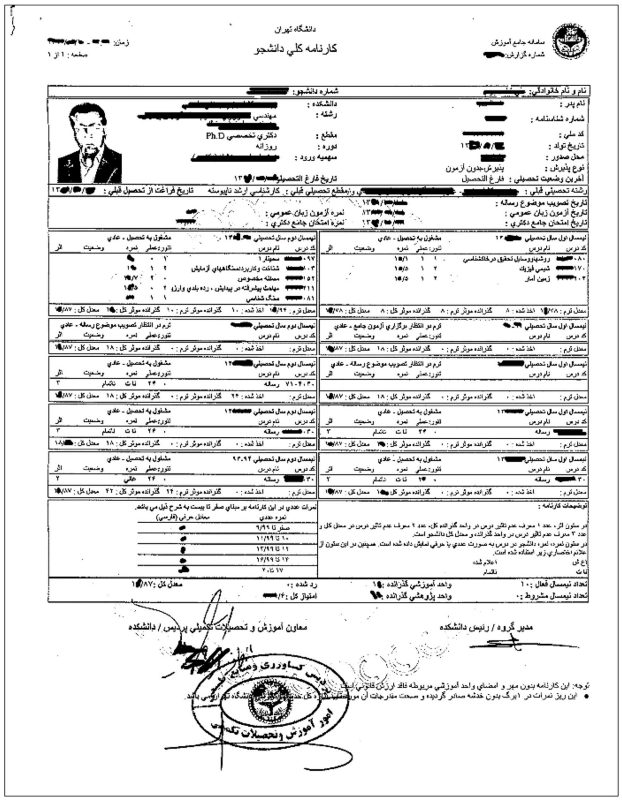


Figure 14. An example of an image with a complex structure.

In this study, 100 simple-structure images [45] and 200 complex-structure images are used as the test set. Since the OCR system generates an output for each set of black pixels identified by the segmentation module, the recognition of images with a complex structure is more difficult than for simple-structure images.

For evaluation, accuracy is used as the criterion to indicate line elimination (the lines that are correctly removed) and line preservation (the lines that correctly remained and are not removed). The line accuracy is calculated using Eq. (12). To calculate the accuracy, the text of the images and the output of the Persian OCR are labelled manually.

$$Accuracy = \frac{\#\ True\ Detected\ Lines}{AllLine} * 100. \qquad (12)$$

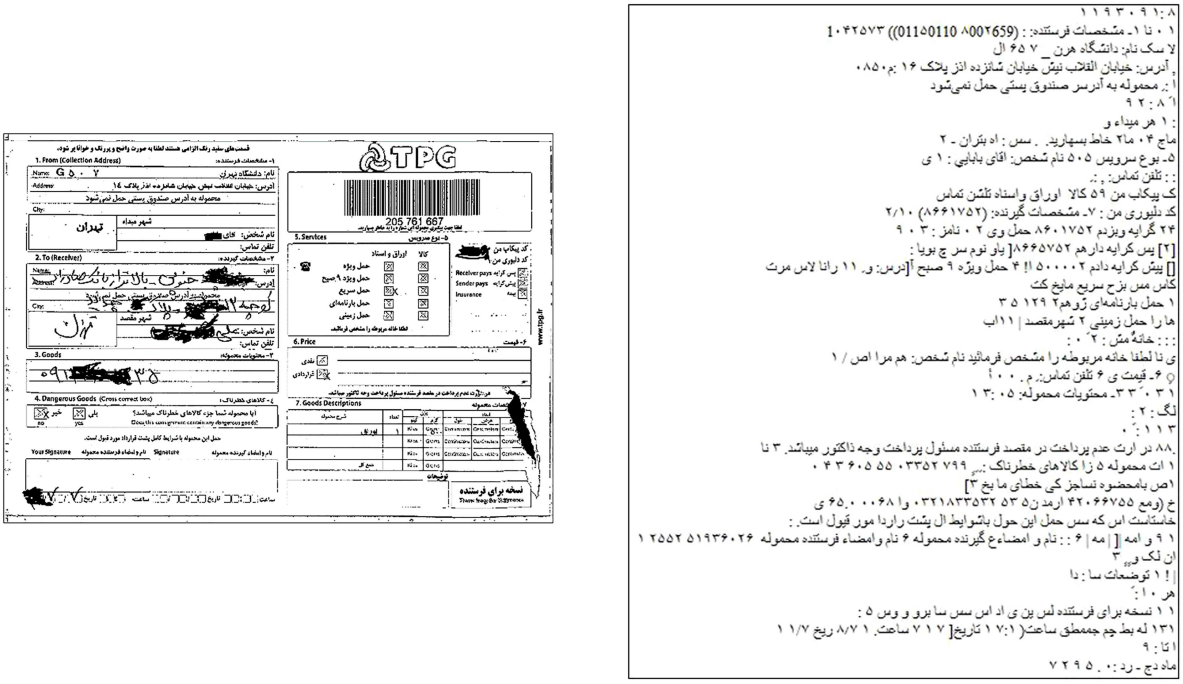
4.3. Results of evaluations

The baseline accuracy (before post-processing) of the Khana Persian OCR for images with a simple structure and complex structure is 98% and 39%, respectively. In addition, the results of the Bina OCR after application on simple- and complex- structure images are 98% and 58%, respectively (see Figure 16). Obviously, these results show the poor performance of the Khana and Bina OCR systems on images with a complex structure.

The reason for the low accuracy (for complex-structure images) is that the output of OCR for this kind of image contains several lines that do not carry any meaning. It means that the lines consist of words with a single letter and many numerals. An example of these lines is demonstrated in Figure 3.

The goal of the proposed method is to improve the accuracy of the OCR results in dealing with complex-structure images. In parallel, it is required to preserve the accuracy of simple- structure images by applying the proposed method. The accuracy of the outputs of the Khana and Bina Persian OCR systems, after the proposed post-processing, on the images with simple and complex structures, are shown in Figure 17. Rules 1 to 5 and Rules 1 to 7 were tested separately, the reason being that Rules 6 and 7 are strongly beneficial for images with a complex structure, but do not show any improvements in images with a simple structure. The reason is that the average length of lines in images with a simple structure is longer and Rule 6 evaluates lines with long length. In addition, a ratio of long words (words with more than two letters) is considered with the length of the line. This means that long lines are expected to have more long words, otherwise, the line is considered as meaningless and is removed.

Lines with a long length that have several single- or double-letter words, or numerals, are recognized as unworthy lines. Rule 7 is considered for lines with a shorter length. If in the lines with short lengths, the count of numerals and short length words are greater than the number of long words, it shows that the line



(a) (b)

Figure 15. (a) An example of a complex-structure image. (b) The corresponding results of the Khana OCR system.

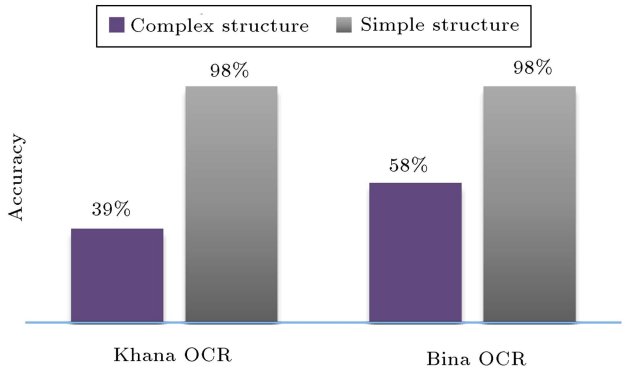


Figure 16. The accuracy of the Khana and Bina Persian OCRs without post-processing.

has meaningless letters and should be eliminated. In this rule, dictionary probability is used to recognize words existing in the dictionary and to protect them against elimination.

In the OCR, most of the black pixels are converted to text. Hence, in the complex- structure images, there are many single-letter and two-letter words in each line that create meaningless lines. However, this challenging issue does not exist in a simple- structure image. Hence, is it concluded that Rules 6 and 7 are appropriate only for images with complex structures and decrease in accuracy in images with a simple structure. As shown in Figure 17, the accuracy of the results in the Khana and Bina OCR systems on images with a complex structure increased from 39% to 93% and 58% to 91%, respectively.

In Figure 18 the effectiveness of each rule is shown separately, to express the effectiveness of each rule and find the most effective. As can be seen, Rule 7 has the greatest and Rule 6 has the lowest impact on performance, respectively. In this evaluation, Rule 1 was applied in all tests.

Figure 19 shows the average accuracy of the results of the Khana and Bina OCR systems after applying post-processing using Rules 1–5 and 1–7 separately on the images with both simple- and complex-structures. As illustrated in Figure 19, after utilizing all rules on the images with a simple structure, the average accuracy is decreased compared to the results without use of the proposed post-processing method (Figure 16). On the contrary, the average accuracy of complex structures is increased to more than 92%.

5. Conclusion and future work

In this paper, the challenges of the existing Persian OCR systems in dealing with images (simple and complex structures) have been evaluated and a rule-based post-processing method has been proposed to address some of these challenges. In order to eliminate meaningless words and lines, the output of the Persian OCR needs to be processed. In the proposed algorithm, five features are extracted in each line and evaluated using seven proposed rules. The proposed rules improve the accuracy of the OCR results by identifying and eliminating meaningless lines. Figure 19 demonstrated the average accuracy in both types of images (images

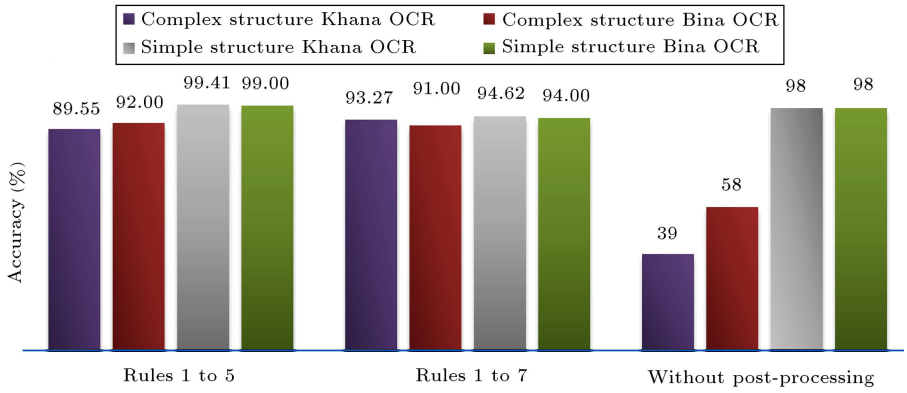


Figure 17. Results of the Khana and Bina Persian OCR systems on the images with simple and complex structures without post-processing and after using the proposed post-processing method for Rules 1-5 and Rules 1-7 separately.

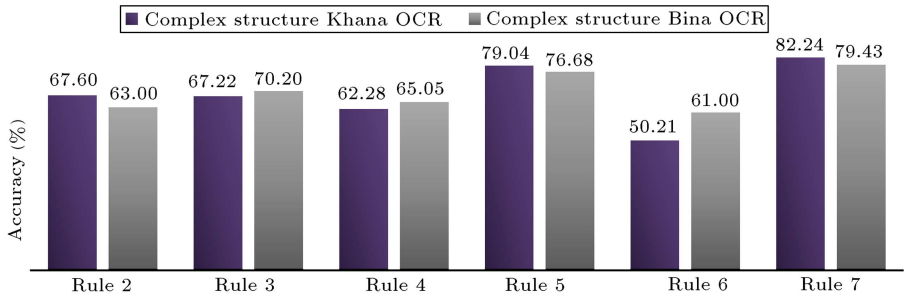


Figure 18. Results of the Khana and Bina Persian OCR systems on the complex- structure images after applying the proposed post-processing approach. The first rule was applied in all tests while other rules (2-7) were applied separately.

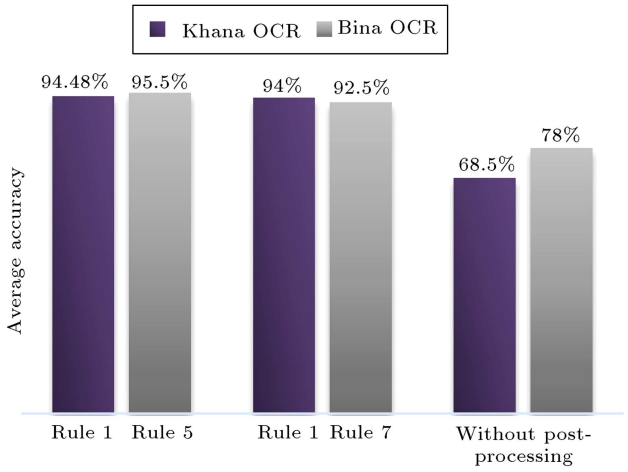


Figure 19. Average accuracy on images with both simple and complex- structures before and after applying the proposed post-processing algorithm. The average accuracy on images after applying Rules 1-5 and Rules 1-7 has been shown separately.

with simple and complex structures) after applying Rules 1–5 and Rules 1–7, compared with the case where post-processing is not present. There are still some challenges in terms of detecting the optimum value of thresholds in each rule that can be analyzed through machine learning techniques in our future work.

References

1. Sajedi, H. "Handwriting recognition of digits, signs, and numerical strings in Persian", *Computers & Electrical Engineering*, **49**, pp. 52–65 (2016).
2. Azadnia, M. "Presenting an expert system for automatic correcting Persian texts", *International Journal of Computer Science and Network Security*, **8**(3), pp. 27–31 (2008).
3. Eikvil, L. *Optical Character Recognition*, Norsk Regnesentral, P.B. 114 Blindern, N-0314 Oslo, (Dec. 1993).
4. Singh, A., Bacchuwar, K., and Bhasin, A. "A survey of OCR applications", *International Journal of Machine Learning and Computing*, **2**(3), pp. 314–318 (2012).
5. Menhaj, M.B. and Adab, M. "Simultaneous segmentation and recognition of Farsi/Latin printed texts with MLP", *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference*, **2** (2002).
6. Raymond, S. "Hybrid page layout analysis via tab-stop detection", *Document Analysis and Recognition, ICDAR'09. 10th International Conference* (2009).
7. Simon, A., Pret, J.-C., and Johnson, A.P. "A fast algorithm for bottom-up document layout analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(3), pp. 273–277 (1997).
8. O'Gorman, L. "The document spectrum for page layout analysis", *IEEE Transactions on Pattern Anal-*

ysis and Machine Intelligence, **15**(11), pp. 1162–1173 (1993).

9. Pritpal, S. and Budhiraja, S. "Feature extraction and classification techniques in OCR systems for handwritten Gurmukhi script-a survey", *International Journal of Engineering Research and Applications (IJERA)*, **1**(4), pp. 1736–1739 (2011).
10. Lehal, G.S. and Singh, C. "A Gurmukhi script recognition system", *Pattern Recognition, Proceedings. 15th International Conference*, **2** (2000).
11. Zand, M., Naghsh Nilchi, A., and Monadjemi, S.A. "Recognition-based segmentation in Persian character recognition", *Proceedings of World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering*, **28** (2008).
12. Khosravi, H. and Kabir, E. "A blackboard approach towards integrated Farsi OCR system", *International Journal of Document Analysis and Recognition (IJDAR)*, **12**(1), pp. 21–32 (2009).
13. Malik, S.A., Maqsood, M., Aadil, F., et al. "An efficient segmentation technique for Urdu optical character recognizer (OCR)", *Advances in Information and Communication*, **70**, pp. 131–141 (2019).
14. Mirzaee, M. "Text detection in images for Persian optical character recognition", MSc Thesis, University Of Tehran, Iran (2012).
15. Ghanbari, N. "A review of research studies on the recognition of Farsi alphabetic and numeric characters in the last decade", *Fundamental Research in Electrical Engineering*, Springer, Singapore, pp. 173–184 (2019).
16. Kameswara Rao, T., Yashwanth Chowdary, K., Koushik Chowdary, I., et al. "Optical character recognition from printed text images", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, **5**, pp. 597–604 (2019).
17. "Bina Persian OCR system", ASR-Gooyesh Co., http://www.binaocr.com.
18. Niwa, H., Kayashima, K., and Shimeki, Y. "Post-processing for character recognition using keyword information", *IAPR Workshop on Machine Vision Applcatron*, Tokyo (1992).
19. Hong, T. "Degraded text recognition using visual and linguistic context", Doctoral Dissertation, University of New York, Buffalo (1996).
20. Kukich, K. "Techniques for automatically correcting words in text", *Acm. Computing Surveys (CSUR)*, **24**(4), pp. 377–439 (1992).
21. Jurafsky, D. and Martin, H.J., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2008).
22. Mays, E., Damerau, F.J., and Mercer, L.R. "Context based spelling correction context-sensitive spell checking based on field association terms dictionaries", *Information Processing & Management*, **27**(5), pp. 517–522 (1991).
23. Beaufort, R. and Thillou, C. "A weighted finite-state framework for correcting errors in natural scene OCR", *Document Analysis and Recognition, ICDAR 2007. Ninth International Conference*, **2**, pp. 889–893 (2007).
24. Bassil, Y. and Alwani, M. "OCR post-processing error correction algorithm using google online spelling suggestion", *Computer Science ArXiv*, **3**(1), pp. 1–9 (2012).
25. Ranka, V., Patil, S., Patni, S., et al. "Automatic table detection and retention from scanned document images via analysis of structural information", *2017 Fourth International Conference on Image Information Processing (ICIIP)*, India (2017).
26. Jahan MAC, A. and Ragel, R. "Locating tables in scanned documents for reconstructing and republishing", *7th International Conference on Information and Automation for Sustainability*, Sri Lanka (2014).
27. Nagata, M. "Japanese OCR error correction using character shape similarity and statistical language model", *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, **2**, pp. 922–928 (1998).
28. Afli, H., Barrault, L., and Schwenk, H. "OCR error correction using statistical machine translation", *International Journal of Computational Linguistics and Applications*, **7**(1), pp. 175–191 (2016).
29. Kesorn, K. and Phawapoothayanchai, P. "Optical character recognition (OCR) enhancement using an approximate string matching technique", *Engineering and Applied Science Research*, **45**(4), pp. 282–289 (2018).
30. Doush, A.I., Alkhateeb, F. and Gharaibeh, H.A. "A novel Arabic OCR post-processing using rule-based and word context techniques", *International Journal on Document Analysis and Recognition (IJDAR)*, **21**(1-2), pp. 77–89 (2018).
31. Magdy, W. and Darwish, K. "Arabic OCR error correction using character segment correction, language modeling, and shallow morphology", *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 408–414 (2006).
32. Ramanan, M., Ramanan, A. and Charles, E.Y.A. "A performance comparison and post-processing error correction technique to OCRs for printed Tamil texts", *Industrial and Information Systems (ICIIS), 2014 9th International Conference*, India (2014).
33. Kolak, O. and Resnik, P. "OCR post-processing for low density languages", *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 867–874 (2005).
34. Zaiz, F., Babahenini, C.M., and Djeffal, A. "Puzzle based system for improving Arabic handwriting recognition", *Engineering Applications of Artificial Intelligence*, **56**, pp. 222–229 (2016).

35. Al-Yousefi, H. and Upda, S.S. "Recognition of Arabic characters", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **8**, pp. 853–857 (1992).

36. Khorsheed, S.M. and Clocksin, F.C., *Structural Features of Cursive Arabic Script*, BMVC (1999).

37. Mahootian, S., *Persian*, Routledge (2002).

38. Awde, N. and Samano, P., *The Arabic Alphabet: How to Read and Write It*, Lyle Stuart (1986).

39. Parhami, B. and Taraghi, M. "Automatic recognition of printed Farsi texts", *Pattern Recognition*, **14**(1-6), pp. 395–403 (1981).

40. Azmi, R. and Kabir, E. "A new segmentation technique for omnifont Farsi text", *Pattern Recognition Letters*, **22**(2), pp. 97–104 (2001).

41. Ebrahimi, A. and Kabir, E. "A pictorial dictionary for printed Farsi subwords", *Pattern Recognition Letters*, **29**(5), pp. 656–663 (2008).

42. Azadnia, M. "Presenting an expert system for automatic correcting Persian texts", *International Journal of Computer Science and Network Security*, **8**(3), pp. 27–31 (2008).

43. Tesseract Open Source OCR engine (main repository), https://github.com/tesseract-OCR/tesseract.

44. Smith, R. "An overview of the Tesseract OCR engine", *9th IEEE Intl. Conf. on Document Analysis and Recognition (ICDAR)* (2007).

45. Persian processing, Tmu-printed-farsi-text-1-100-pp, http://farsiocr.ir/.

Biographies

Zohreh Khosrobeigi received an MS degree in IT at the Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran, in 2016. Currently, she is a PhD student at the School of Computer Science and Statistics, Trinity College Dublin, Ireland. Her main interests are deep learning, NLP, and image processing.

Hadi Veisi is Assistant Professor of Computer Science at the Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran. He is also Head of the Digital Signal Processing Lab. He received his PhD degree in Computer Engineering at Sharif University of Technology, Tehran, Iran, in 2011. His current research includes digital signal processing, speech processing (speech recognition, speech enhancement, speech synthesis, speaker recognition), natural language processing, artificial neural network, and deep learning.

Hamid Reza Ahmadi received BS, MS, and PhD degrees in Electrical and Computer Engineering from the University of Tehran, Tehran, Iran, in 1998, 2001, and 2011, respectively. Since 2012, he has been at the University of Tehran, where he is Assistant Professor with the Faculty of New Sciences and Technologies. His current research interests include data and network security, Software-Defined Networks (SDN), blockchain and DLT, watermarking, and video compression.

Hanieh Shabanian is pursuing a PhD degree in the Department of Electrical and Computer Engineering at The University of Memphis. Her research interests and dissertation projects are at the intersection of ocular imaging, computer vision, image processing, and machine learning. Ms. Shabanian is building a prototype imaging system for three-dimensional imaging of ocular structures in the Computational Ocularscience Laboratory at The University of Memphis. A secondary focus of her research work is on developing optical biomarkers of retinal diseases.