Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 1, 2025 Serial no. 111
DOI: 10.22034/tjee.2024.62945.4877

# Optical Character Recognition (OCR) in Cursive Scripts Using Object Detection Networks

Mojtaba Gandomkar *, Sahar Khoramipour

Department of Electrical and Computer Engineering, Jundi-Shapur University of Technology, Dezful, Iran
E-mail addresses: gandomkar@jsu.ac.ir, saharkhoramipour17@gmail.com
*Corresponding author

Received:13/08/2024, Revised: 22/10/2024, Accepted:04/12/2024.

**Abstract**

Optical Character Recognition (OCR) in cursive scripts, where the letters of a word are joined in a flowing manner and overlap in both directions, deals with the struggles raised while segmentation of unrecognized characters and recognition of unseparated characters. In this paper, we propose using object detection models for character detection in cursive scripts. Simplicity of implementation and efficiency of this method in recognition of handwriting-style fonts are investigated and discussed. Here, YOLO model is used to separate and classify the characters of arbitrary three-letter words in Persian script as a case study. Initially, we generated synthetic datasets suitable for the YOLO network from handwriting-style Persian fonts, such as Maneli and IranNastaliq. By using the YOLO model, we achieved high Precision of 98.5% in character detection of Maneli font and 97.6% for a mixture of words in Maneli and IranNastaliq fonts, while the accuracy for the regular font Arial was almost 100%. Then, we challenged the proposed model by adding noise, blur, and skewness to the samples. Furthermore, we utilized a multi-layer perceptron (MLP) model to predict the words from the characters detected and localized by YOLO with the accuracy of 99.8% for Maneli font and 97.7% for a mixture of words in Maneli and IranNastaliq fonts, while the word detection accuracy for the regular font Arial was almost 100%. This approach enables us to recognize complete words accurately from complex handwriting-style fonts, without using a Persian vocabulary dictionary.

**Keywords**

Optical character recognition (OCR), Object detection, YOLO model, Multi-layer perceptron (MLP), Persian script, Handwriting-style fonts.

## 1. Introduction

Digitization of texts has provided countless opportunities for editing, searching, classifying, analyzing, transmitting and compilation of human knowledge. However, a large amount of texts are still produced or stored on papers or through images and a vast amount of information is still manually processed. In this regard, Optical Character Recognition (OCR) technics are developed for conversion of images containing printed or handwritten text with the script of any language into a machine-readable text. OCR has gained significant attention in applications related to business printed documents [1], historic books or newspapers [2], handwritten documents [3], and even detection of texts in videos and pictures captured by camera [4].

Each OCR system consists of several modules. As shown in Fig. 1, first, the document is imported to the system as images with an appropriate resolution. Then, a preprocessing module removes noise and deals with dimensional deviations and quality distortions such as skewness, rotation, curved baseline, blurring, low-contrast imaging, and the effects of ambient light and shadows which can be the main obstacles for the accuracy of OCR. In the next stage, textual and non-textual areas of complex images are distinguished using simple text region detection tools or more powerful Document Layout Analysis (DLA) tools such as Layout Parsers based on deep learning [5]. Then, an internal line division module divides the textual area into the lines and the lines into the word regions. To detect the characters of the words, the word region may be segmented into letter or subword regions or passed directly to the feature extraction module. Feature extraction module generates determinative features of the word, subword, or the separated letter and passes them to the recognition module to recognize the characters. According to the characters recognized in the word region, in the text recognition module, the word is guessed directly or with comparison to the words of a corresponding dictionary. Separation of the letters (segmentation), feature extraction and character recognition stages are very dependent on the language and its script and efforts to find faster and more accurate methods for diverse languages continue [6-8].

In some scripts and formal fonts across different languages, characters and letters are written as block letters in which the letters are not joined or the letters are connected, but in a way that they do not overlap in one direction [9]. This characteristic is utilized during segmentation to separate the letters easier. However, many people around the world use cursive scripts in which the letters of a word are joined in a flowing manner. Form of the letters may change according to their position within the other letters of the word. The main challenge in cursive scripts is separation of the letters within the words. Additionally, character recognition for these scripts needs more complex processes to identify different forms of a letter. Segmentation and character recognition from handwritten documents and documents containing handwriting-style fonts is involved more with the issue of joint letters that overlap in two directions. We are going to deal with these challenges, with concentration on Persian script as a case study.

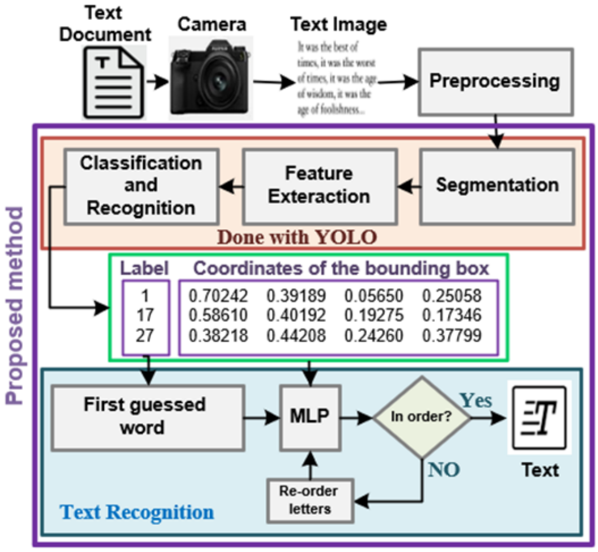


Fig. 1. Procedure of the proposed method for recognizing the letters in images using the YOLO network and recognizing the text (word) using the MLP network.

The Persian script is a cursive script also known as the Perso-Arabic script [10]. This script is mostly written from right to left. However, numbers, mathematical expressions, and numeric dates are embedded from left to right. As depicted in Table I, Persian alphabet has 32 main letters (“آ” is counted with “ا”) that do not have capital form but every letter has different forms according to the position of the letter in the word. The letters “آ, ا, د, ذ, ر, ز, ژ, و” have two forms (final, and independent) and the other letters appear in four forms (initial, medial, final, and independent). For example the letter “س” (Sin), is in initial form in “سبز” (green), in medial form in “هست” (exists), in final form in “حس” (sense) and in independent form in “حواس” (senses). In addition, there are groups of letters that have the same basic structure but they differ only in the number of dots and the position of the dots placed over or under the letter such as “ث, ت, ب, پ, and ی, ن in initial and medial forms”, “ج, چ, ح, خ”, “د, ذ”, “ر, ز, ژ”, “س, ش”, “ص, ض”, “ط, ظ”, “ع, غ”, or in diacritics such as “ک, گ”. Table I specifies the diversity of different letter forms in Persian script and the class number of each letter.

Basically, in a cursive scripts like Persian/Arabic script, separation of letters and recognition of characters are two intertwined processes. In classical methods, reliance is primarily placed on explicit features such as geometric and correlation-based ones. Azmi et al. presented an algorithm for segmentation based on conditional labeling of upper contours. Additionally, they utilized a preprocessing technique that adjusts the local baseline for each subword [11]. Khosravi et al. proposed a unified OCR system for Persian text. This system utilizes information from multiple Knowledge Sources (KS) and manages them in a blackboard approach. The system was able to recognize ten popular Farsi fonts using a Font Recognizer module [12]. Hajihashemi et al. utilized a Holography graph neuron system to memorize patterns of separated Persian/Arabic characters. The designed architecture was robust against noise, and capable in recognizing patterns of separated Persian characters effectively [13]. Qods and Sohrabi utilized a Hidden Markov Model (HMM) for identifying the main body of characters. Their final detection relies on delayed strokes (dots and small marks) and hidden Markov models [14].

**Table I.** Letter forms in Persian script and the class number of each letter.

| Class number | Character name | Independent | Initial | Medial | Final |
|---|---|---|---|---|---|
| 0 | Aaa | آ | - | - | ـآ |
| 1 | Alf | ا | - | - | ـا |
| 2 | Beh | ب | بـ | ـبـ | ـب |
| 3 | Peh | پ | پـ | ـپـ | ـپ |
| 4 | Teh | ت | تـ | ـتـ | ـت |
| 5 | Seh | ث | ثـ | ـثـ | ـث |
| 6 | Jim | ج | جـ | ـجـ | ـج |
| 7 | Che | چ | چـ | ـچـ | ـچ |
| 8 | Hej | ح | حـ | ـحـ | ـح |
| 9 | Khe | خ | خـ | ـخـ | ـخ |
| 10 | Dal | د | - | - | ـد |
| 11 | Zal | ذ | - | - | ـذ |
| 12 | Reh | ر | - | - | ـر |
| 13 | Zeh | ز | - | - | ـز |
| 14 | Zhe | ژ | - | - | ـژ |
| 15 | Sin | س | سـ | ـسـ | ـس |
| 16 | Shin | ش | شـ | ـشـ | ـش |
| 17 | Sad | ص | صـ | ـصـ | ـص |
| 18 | Zad | ض | ضـ | ـضـ | ـض |
| 19 | Taa | ط | طـ | ـطـ | ـط |
| 20 | Zaa | ظ | ظـ | ـظـ | ـظ |
| 21 | Ein | ع | عـ | ـعـ | ـع |
| 22 | Ghein | غ | غـ | ـغـ | ـغ |
| 23 | Feh | ف | فـ | ـفـ | ـف |
| 24 | Ghaf | ق | قـ | ـقـ | ـق |
| 25 | Kaf | ک | کـ | ـکـ | ـک |
| 26 | Gaf | گ | گـ | ـگـ | ـگ |
| 27 | Lam | ل | لـ | ـلـ | ـل |
| 28 | Mim | م | مـ | ـمـ | ـم |
| 29 | Nun | ن | نـ | ـنـ | ـن |
| 30 | Vav | و | - | - | ـو |
| 31 | Heh | ه | هـ | ـهـ | ـه |
| 32 | Yeh | ی | یـ | ـیـ | ـی |

Sadri et al. utilized the histogram of gradients method, Multi-Layer Perceptron (MLP) neural networks, and nearest neighbor classifiers [15]. Khorashadizadeh et al. constructed a model by combining four directional Chain Code Histogram (CCH) and Histogram of Oriented Gradients (HOG). To achieve a higher detection rate, they extracted local features at two levels with 2×2 and 1×1 grids [16]. Parseh et al. utilized combined features including segmentation, intersection count, and size of holes, and used Principal Component Analysis (PCA) to reduce feature space dimensions [17]. Montazer et al. utilized a fuzzy neural inference engine to detect Persian numeric characters. This engine adopts a holistic approach to character recognition by comparing the features of unknown characters with the features of existing characters, which are determined through a fuzzy inference engine based on specific fuzzy rules [18]. Pourreza et al. suggested addressing the challenges of Persian OCR by focusing on detecting subwords instead of individual letters and employing comprehensive subword vocabularies [19]. Aghbari et al. divided the handwritten document image into words and further segmented each word into its constituent subword parts. Then, they extracted several structural and statistical features from these segments. Finally, they employed a neural network to classify input vectors into word classes [20]. However, detection of subwords leads to high computational requirements due to the actual number of subwords in the dictionary. Working on character level requires significantly lower data than what that is required for recognition on the word or subword level.

In the process of separation and recognition of the letters, artificial intelligence and the algorithms based on Artificial Neural Networks (ANN) have been very helpful. Nanehkaran et al. evaluated the performance of traditional methods such as K-nearest neighbor, ANN and Support Vector Machine (SVM) in recognizing digits. The ANN offered better execution time than the SVM, but its accuracy was lower [21]. Khorashadizadeh et al. used a SVM with a radial basis function kernel for classification to achieve high accuracy in their classical methods [16]. Parseh et al. integrated a non-linear multi-class SVM classifier instead of a last fully connected layer in their Convolutional Neural Network (CNN) structure [22].

Traditional neural networks relied heavily on handcrafted features and template matching techniques, which often failed to generalize to diverse datasets and complex scripts, such as cursive writing. In recent years, deep learning concept has revolutionized OCR systems by using complex and deep neural networks that gives the ability to automatically learn relevant features from raw data [23, 24]. In this regard, Bonyani et al. used various architectures of deep neural networks such as DenseNet, Xception, ResNet50, and VGG16 to recognize Persian letters, numbers, and words [9]. Khosravi et al. utilized an optimized LeNet network for recognizing Persian handwriting using meta-heuristic training [25]. Tesseract is another deep learning model widely used for OCR of complex scripts [26].

To create the word from the recognized letters (Text Recognition), most approaches in the literature rely on vocabulary-based methods. Recurrent Neural Networks (RNNs) [27] especially Long Short-Term Memory (LSTM) networks [27, 28] are designed to store useful information in sequential data due to their internal memory. Gadikolaie et al. presented a segmentation-based approach for offline recognition of handwritten Persian words, where each word was divided into subwords, followed by an RNN with True/False outputs for each subword [29]. Given the sequential nature of textual data, some researchers have utilized LSTM for designing a better OCR system [30], which can also be applied in Persian OCR. BLSTM network is a modified version of LSTM that can simultaneously get information from past and future contexts [31].

Recently, the concept of object detection is strongly improved by the deep learning algorithms. In this concept, several objects in different classes are detected within an image, and the corresponding region or even the corresponding pixels are determined. In other words, segmentation, localization, feature extraction and recognition processes are done simultaneously for many objects that seems to be perfectly matched to the requirement of separation and recognition of characters in cursive scripts. Object detection for OCR is investigated with CNN based models such as faster R-CNN (Region based CNN) that are deficient in terms of speed and the number of recognizable object forms [32]. SSD (Single Shot Detection) and YOLO (You Only Look Once) are two famous object detection networks that can rapidly detect multiple small objects of several classes in an image. Yolo is recently used for OCR purposes [33-35].

In this paper, we are going to propose a Persian text detection technique based on the YOLO object detection model that may be easily adapted to the script of other languages. In this manner, as shown in Fig. 1, character segmentation (localization) and classification of that character (identification) are performed simultaneously inside a word region. This approach avoids the struggles raised while segmentation of unrecognized characters or recognition of unseparated characters within a word region. Different forms of a letter in various fonts can be recognized if appropriate training data is provided. YOLO is previously used for detection of textual regions [36] but the idea of using YOLO for character separation and recognition is the innovation of this work. In addition, we designed a simple Multi-Layer Perceptron (MLP) to rearrange the letters detected by the YOLO network for word detection or text detection. In this way, based on the detected letters by the YOLO network and the corresponding bounding box coordinates, the MLP network labels the possible letter pairs with two classes, “in order” and “not in order”. Therefore, the desired word is easily detected by putting the detected letters in order. In this article, we generate our own synthetic datasets to be compatible for YOLO.

The organization of this paper is as follows. In Section 2, the dataset generation method is explained. Next, in Section 3, the method of recognizing letters in an image with the YOLO network is described, and then, Section 4 describes the results for letters and word detection. Finally, conclusions and future work are presented in Section 5.

## 2. Generation of the datasets

Deep neural network models require large-scale training datasets to achieve high accuracy in segmentation and recognition. Generally, datasets are classified into real and synthetic types due to the method used for data collection. Real datasets for OCR are created by scanning documents such as newspapers or scene images [37]. HODA, Sadri, and IranShahr are some real datasets in Persian OCR [9, 17]. To create synthetic datasets, images of characters or words are randomly generated using text and image editors with various fonts and sometimes in different backgrounds. In synthetic datasets, there is no restriction on the number of samples and many arbitrary parameters can be considered in the investigations.

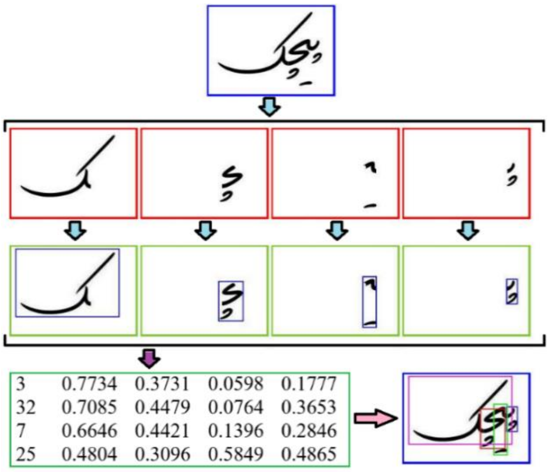
In this article, a synthetic dataset is generated by developing a personal code to demonstrate the ability of the proposed object detection concept of YOLO for OCR of cursive scripts like Persian script. To train a YOLO network, a dataset of images is required with a text file for each image that contains the label of the characters existing in the image and the corresponding coordinates of bounding box of each character. In this paper, we decided to generate our synthetic dataset. To challenge our proposed method, we use handwritten-style Maneli and IranNastaliq Persian fonts because of their resemblance to human handwriting and their capability of selecting alternative letter forms by adding some special characters before or after a letter.

Generation of the dataset requires minimal user intervention and is accomplished only by providing a set of words and determining the font size, font name and image size. In this manner, the code will be extendable for new fonts and even for languages other than Persian with simple modifications. Different forms of a letter in a word (initial, medial, final, and independent), as well as the alternative forms revealed by combination of the letter with special characters, are all labeled with the original letter name. This process challenges the YOLO network to identify the letters, but in return, simplifies the procedure and demonstrates the advantages of the proposed idea.

There is no restriction on the number of letters in each word. However, a dataset consisting of three-letter words has been primarily considered. The three letters of a word are randomly selected from the alphabet list. Therefore, these words cover almost all different forms of a letter (initial, medial, final, and independent) with an almost uniform distribution. Training of the YOLO network with three-letter words is almost adequate for detection of letters of words with more than three letters. The three-letter words are not necessarily meaningful in Persian. Therefore, this approach is not restricted to recognition of the words of Persian language vocabulary. We will show that character detection in this method does not require a huge number of training samples even for handwritten-style fonts (less than a few thousand images of words consisting of three letters).

The dataset generation process is shown in Fig. 2. The process of generation of data files for YOLO network starts by writing the desired word in black on a white background using a text editor. The image of this word is then saved as the original image. In the next step, the color of the written word is changed to white, but each letter is individually made black to generate a new image. In the new image, the shape and position of the letter in the word are preserved. In this manner, the position of the letter in the word is easily recognizable even if overlapping with the positions of other letters. Finally, the original image of the word along with a text file containing the coordinates of the bounding box of each letter is reported for training the YOLO network. In the text file, after mentioning the label of each letter, the location of this letter in the original image is indicated with four numbers. These numbers are, in order, the relative position of the letter's center point horizontally, the relative position of the letter's center point vertically, the relative width of the letter horizontally, and the relative height of the letter vertically.

As is shown in Fig. 2, the text file is generated for an image containing the word "پیچک" (ivy) in Maneli font. This word has four letters with the labels 3, 32, 7, and 25, respectively. As it can be seen, the letters of this word spatially overlap both horizontally and vertically in Maneli font. If we consider the letter "چ" as an example, it spatially overlaps with the letters "ی" and "ک". Furthermore, the rectangular area that includes all the points of "ک" will also include a portion of "پ", "چ", and "ی" within itself.



|  |  |  |  |  |
|---|---|---|---|---|
| 3 | 0.7734 | 0.3731 | 0.0598 | 0.1777 |
| 32 | 0.7085 | 0.4479 | 0.0764 | 0.3653 |
| 7 | 0.6646 | 0.4421 | 0.1396 | 0.2846 |
| 25 | 0.4804 | 0.3096 | 0.5849 | 0.4865 |

Fig. 2. Dataset generation procedure. A text file containing the labels of the letters of "پیچک" (ivy) and the corresponding bounding box coordination is generated for training YOLO network (with Maneli font).

These mentioned steps are followed correctly for most words. However, when it comes to the ligatures (specific combination of letters like "ل" (Lam) and "ا" (Alf) that changes into "لا" (La)), as they are drawn as a continuous single object, most text editors do not allow setting the color of single letters (i.e. "ل" or "ا") separately. This limitation causes difficulty in spatial separation of single letters of ligatures. To address this issue, the ligatures of each font should be extracted, and the positions of ligatured letters need to be adjusted manually. For instance, combinations like "لا", "کا", "گل", and "گلا" in Maneli font require position adjustments. Although this process requires manual adjustments for each font, doing it once is enough forever. We do not have restrictions on the number of images, and all steps are performed automatically without user intervention.

In this paper, we have generated three different datasets. Each dataset consists of images of three-letter words with a resolution of 300 dpi (image size of approximately 640×512 pixels). The first dataset consists of 1321 images of words in Maneli font. In the second dataset, we added 323 samples with various alternative letter forms to the first dataset. This alternative letter forms are created by adding special characters, such as "أ", "ء", "ؤ", "ئ" in Maneli font or "Tatvil" character in IranNastaliq, before or after the main letter. These samples bring the text closer to human handwriting and challenge the YOLO model in character detection. Ultimately, the second dataset consists of 1644 images in Maneli font. In the third dataset, we added 1014 samples of three-letter words in IranNastaliq font to the second dataset. The third dataset comprising 2658 images with Maneli and IranNastaliq fonts. For each image, a text file has been automatically generated, containing the positions of each letter and their labels. Figs. 3 and 4 show the variety of different forms of letters in some words with Maneli font and IranNastaliq, respectively. Bounding boxes are depicted around every letter.

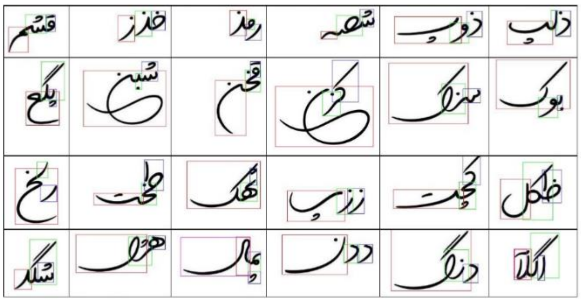


Fig. 3. The variety of different forms of letters in some words with Maneli font.

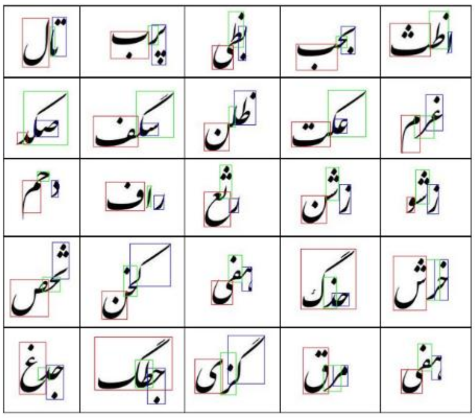


Fig. 4. The variety of different forms of letters in some words with IranNastaliq font.

## 3. Methods

### 3.1. *YOLO algorithm*

YOLO is a convolutional neural network specialized for object detection that extracts features from images with convolutional layers and predicts bounding boxes of multiple objects using fully connected layers. This network is implemented using the DarkNet framework and provides a unified network that can be used for both object classification and localization. Additionally, the likelihood of each region is also estimated [33-35]. In this paper, we are going to propose the use of object detection capability of YOLO to build an OCR system for cursive scripts such as Persian script as a case study. We built our object detection model using three personalized synthetic datasets described in Section 2.

YOLO was first introduced in 2016 and quickly outperformed other object detection methods [38]. In the following years, scholars have published several YOLO subsequent versions [39]. We train a custom object detection model using a pre-trained YOLOv5 to recognize relevant objects (characters) in the images. Architecturally, YOLOv5 has been implemented in PyTorch, making it faster and lighter in comparison to previous versions. Among the various versions of YOLOv5, we use the YOLOv5m6 model, which was more compatible with our datasets and yielded better results in tolerable times. Other versions of YOLO or future more advanced object detection models may be required for huge datasets containing samples in more fonts, more languages and even handwriting documents.

### 3.2. *Performance criteria and evaluation*

Initially, we divide the custom dataset into 80% for network training and 20% for network validation. We chose a batch size of 16 and conducted the training process for 100 epochs. Finally, for network evaluation, we used criteria such as precision, recall, F1-score, and mAP. The formulas used to determine these criteria are listed as below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \tag{4}$$

Here, true positive (the number of objects accurately identified as positive) is denoted by TP, false positive (the number of objects predicted to be positive but were actually negative) by FP, true negative (the number of objects correctly predicted as negative) by TN, and false negative (the number of objects mistakenly predicted as negative) by FN. Additionally, mAP stands for Mean Average Precision, serving as an evaluation metric for the performance of pattern detection and classification models. mAP indicates the average precision across all classes or different categories present in the dataset, where *n* represents the number of classes, and $AP_k$ represents the average precision of the class number *k*.

## 4. Results

### 4.1. *Applying YOLO on undistorted samples*

In the initial stage, we focus on detecting and separating the constituent letters of a word by applying the YOLO network on our synthetic dataset. We evaluate the Precision (accuracy of the positive predictions made by the model) and Recall (ability to correctly identify all positive instances in the dataset) and F1-Score (measures the overall performance of the network in detecting specific patterns and categories) metrics, along with two important metrics, mAP50 and mAP50-95. mAP50 refers to the mean average precision for the top 50% of results. This metric indicates the model's accuracy in predicting categories with the highest scores or potentially more important categories. mAP50-95, represents the mean average precision for the probability range of 50% to 95% of the top results. This metric reflects the model's accuracy in predicting categories with various and extensive probabilities and scores.

Fig. 5 illustrates the results obtained for the three aforementioned datasets. The YOLOv5 network has successfully detected and separated letters in all datasets. It can be observed that with the first dataset, the evaluation metrics of Precision, Recall, F1-score, mAP50, and mAP50-95 are 97.6%, 96.9%, 97.2%, 98.7%, and 95.2%, respectively and the letters are successfully identified. With the second dataset, there was a slight increase in all evaluation metrics for letter separation.

Although we have added more alternative character forms to the second dataset in comparison to the first dataset, more regular characters have been added too, which has increased the Precision. Consequently, with the second dataset, the evaluation metrics Precision, Recall, F1-score, mAP50, and mAP50-95 reached values of 98.5%, 98.4%, 98.4%, 99.2%, and 96%, respectively. Additionally, with the third dataset, consisting a new font, less than 1% decrease in the evaluation metrics is seen, demonstrating the network's stability in recognizing letters with different fonts. Consequently, it is observed that with the third dataset, the evaluation metrics Precision, Recall, F1-score, mAP50, and mAP50-95 reached values of 96.9%, 96.3%, 96.5%, 98.4%, and 92.4%, respectively. In Fig. 6, we see the output results of YOLO on some three-letter word images containing bounding boxes around the letters of the words.

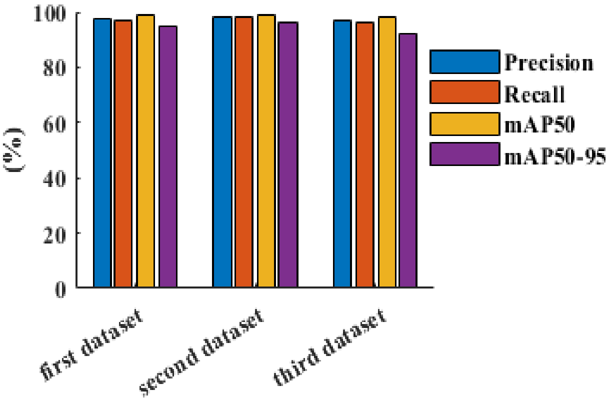


Fig. 5. Evaluating the accuracy of the YOLO network on three different datasets.

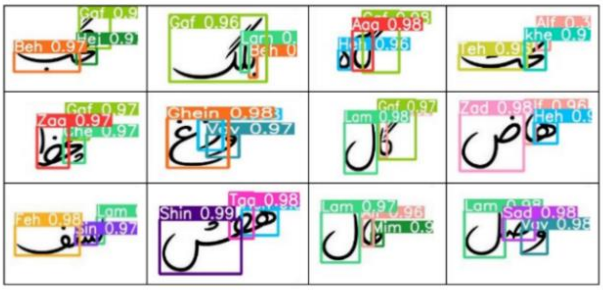


Fig. 6. Samples of results of the YOLO network for detection of letters.

Fig. 7 illustrates the Confusion Matrix for detection and classification of characters with the first dataset. The YOLO network successfully separated and identified characters. As mentioned earlier, there are some challenges in Persian script that had concerned us about the network's ability to correctly identify the characters, but the network has overcome them all. For example, the model has recognized the letters with similar shapes such as “د, ذ, ر, ز, ژ, و” truly, or the letters in specific ligatures such as “ل” (Lam) and “ا” (Alf) forming “لا” (La) are recognized. In addition, the network has been able to accurately separate and identify the overlapping letters in handwriting-style fonts. It is worth mentioning that the network only struggles in detecting the background pixels, which can be attributed to the mixed nature of letters in a word.

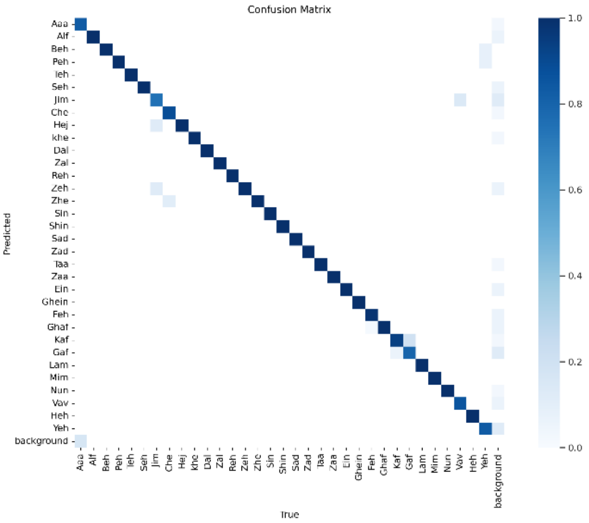


Fig. 7. Confusion Matrix diagram for detection of letters in a word by YOLO.

### *4.2. Adding noise, blur and skew to the samples*

In the next stage, to challenge the network's detection capability, we add artificial disruptions such as noise, blur, and skew to the dataset images and evaluate the effect of these disruptions on the detection precision. Various conditions are examined to observe the extent to which the network can detect the letters truly.

Fig. 8 displays images with various artificial distortions applied with different intensities. Noise is an important challenge in OCR systems. Salt-and-Pepper noise is one of the noise types that regularly affects the quality of the captured images. We introduced artificially generated Salt-and-Pepper noise to the original images of the first dataset at varying intensities, ranging from low noise (level 0.2) to high noise (level 0.8), which completely distorts the image. Another challenge in OCR is detecting text from a blurry image that users may inadvertently create using their phone cameras. To simulate this scenario, we introduced various degrees of blurriness (1, 10, 15, and 20) to the images in first dataset. Another disruption we considered for the images is the skew. Here, first we define a max skew level, $S_{\max}$. Then, all images are randomly skewed using the linear transformation matrix {1 $s_1$; $s_2$ 1}, where $s_1$ and $s_2$ are random numbers in the range of $-S_{\max}$ to $+S_{\max}$. This skews the images randomly in different directions. Bounding boxes must be updated when the image is skewed.

As shown in Table II, the YOLO network has performed well under various conditions and has been able to detect and recognize the letters with high precision. An interesting observation in this experiment is that in specific conditions where letters are indiscernible to the human eye, such as where the noise level is 0.8, the excessive blur level is 20, or the skew level is 0.5, the YOLO network has succeeded in detecting letters with relatively good accuracy even in these challenging conditions.

Now, we intend to challenge the YOLO network further by combining these distortions. According to Fig. 9, we use the first dataset as the basis and first add Salt-and-Pepper noise to all its images. In the next step, we will blur 50% of the images and skew the remaining 50%. This experiment will be repeated in the following

scenarios with different intensities. Distortion level for each image is randomly applied from 0 to the upper value specified as:
Scenario #1: Noise level (0 to 0.2), blur level (0 to 4), skew level (0 to 0.4),
Scenario #2: Noise level (0 to 0.4), blur level (0 to 6), skew level (0 to 0.5),
Scenario #3: Noise level (0 to 0.6), blur level (0 to 8), skew level (0 to 0.8).

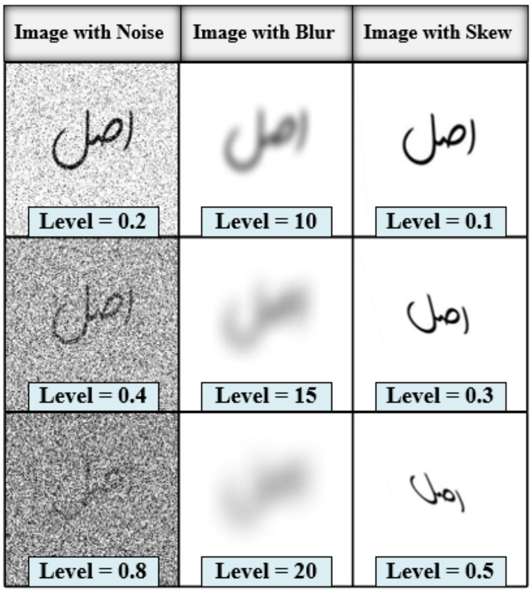


Fig. 8. Samples showing the effect of artificial distortions applied to the images.

**Table II.** The impact of artificial disruptions on the accuracy of the YOLO network in letter detection.

| Artificial Disruptions | Level | Precision (%) | Recall (%) | F1-Score (%) | mAP50 (%) | mAP50-95 (%) |
|---|---|---|---|---|---|---|
| Salt-and-Pepper Noise | 0.2 | 97.1 | 97.8 | 97.4 | 99.3 | 94.8 |
| | 0.4 | 97.1 | 98.5 | 97.7 | 99.3 | 94.4 |
| | 0.6 | 94.5 | 94.7 | 94.5 | 97.8 | 89.3 |
| | 0.8 | 75.6 | 70.8 | 73.1 | 79.5 | 62.5 |
| Blur | 1 | 97.2 | 98.2 | 97.6 | 99.2 | 96.6 |
| | 10 | 93.3 | 94.3 | 93.7 | 96.4 | 90.5 |
| | 15 | 84.4 | 73.4 | 78.5 | 84.8 | 71.1 |
| | 20 | 77.8 | 64.7 | 70.6 | 76.1 | 59.5 |
| Skew | 0.1 | 96.2 | 97.4 | 96.7 | 99.1 | 87.0 |
| | 0.3 | 96.4 | 96.2 | 96.2 | 99.5 | 81.9 |
| | 0.5 | 95.0 | 95.9 | 95.4 | 97.7 | 84.1 |

In this experiment, we increase the intensity of artificial disturbances at each stage, challenging the YOLO network further. As seen in Table III, we observed a slight decrease in network accuracy at each stage. However, the network successfully overcomes this challenge and detects the letters with still good accuracies. Based on the results, it can confidently be stated that the YOLO network has the ability to separate characters in various harsh conditions, including samples from regular and alternative characters forms of Maneli font and the characters of IranNastaliq font, as well as in conditions such as noisy, blurred, and skewed samples. This method can address many of the challenges in recognizing the characters of cursive scripts worldwide.

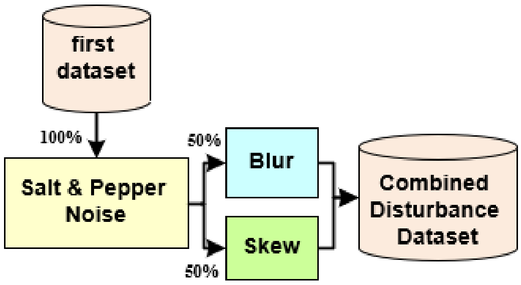


Fig. 9. Combining various types of disturbances with different intensities.

**Table III.** The results of the YOLO network under different combinations of disturbances with varying intensities.

| Scenarios | Precision (%) | Recall (%) | F1-Score (%) | mAP50 (%) | mAP50-95 (%) |
|---|---|---|---|---|---|
| **#1** | 95.2 | 96.5 | 95.8 | 98.9 | 84.6 |
| **#2** | 94.8 | 91.5 | 93.1 | 96.8 | 80.0 |
| **#3** | 91.5 | 85.0 | 88.1 | 91.6 | 73.7 |

### *4.3. Word detection algorithm*

Output of an object detection model is the label of letters and their bounding box. Therefore, another step is required to rearrange the letters and guess the word. As shown in Fig. 1, we trained an MLP neural network to rearrange the word letters using the localization information of the detected characters. The layers used in this network include feature input layer, fully connected layer, batch normalization layer, ReLu layer, fully connected layer, SoftMax layer, and classification layer. We used the Adam optimizer for training this network.

We use the label and the coordinates of the corresponding bonding box of each letter pair as features to train the MLP model with two “in order” and “not in order” classes. For instance, as illustrated before in Fig. 2 (section 2), output of YOLO for the word “پیچک” has four rows of data beginning with the labels 3 (“پ”), 32 (“ی”), 7 (“چ”), and 25 (“ک”). The letter pairs {3, 32}, {3, 7}, {3, 25}, {32, 7}, {32, 25}, and {7, 25} are in order and if we reverse this letter pairs as {32, 3}, {7, 3}, {25, 3}, {7, 32}, {25, 32}, and {25, 7}, they are not in order. The word detection model starts with a random word containing all the detected letters by YOLO, then checks if the letters in this word are “in order” with their next letter. If a letter and its next letter are “not in order”, their place in the word is replaced and this process is continued to reach a word with all letters “in order”.

Every three-letter word has 6 letter pairs (3 “in order” and 3 “not in order”). First, we use the true results of YOLO on the images of the first dataset to train the word detection model. To test the word detection model, first we applied it on 940 images with Maneli font and three-letter words different from the words of the first dataset. The accuracy of word detection model (number of truly detected words / total number of the words) is about 99.8%. Next, we trained the MLP model with the true results of YOLO on the images of the third dataset (containing samples with plain character forms of Maneli font, alternative character forms of Maneli, and character forms of IranNastaliq). Then the model is tested on new images different from the words of the third dataset (containing 617 samples with plain character forms of Maneli font, 365 alternative characters of Maneli, and 490 character forms of IranNastaliq). Name of the font of the images is intentionally not handed to the MLP model as a feature. The accuracy of word detection reduced slightly to 97.7%. This experiment shows that the MLP model can rearrange the letters detected by YOLO of at least two fonts (without knowing the font name of the image) with a good accuracy.

Word detection in our proposed approach does not rely on vocabularies [39] and only the localization information of characters are used to guess the sequence of recognized characters in the word. However, use of vocabularies or use of LSTM may be helpful to increase the word detection accuracy.

### *4.4. Comparison with other works*

In Table IV, we present a comparison of our work with the closest works that attempted object detection for OCR. Here, CRR is character recognition ratio and WRR is word recognition ratio. Our work is more extendible according the method we propose for sample generation. The accuracy that we have achieved is interesting according to the cursive script and the handwriting-style fonts we have investigated.

In addition, we compare our work with the results of two advanced online OCR services that support Persian language with high quality (Google’s OCR and OLOCR.com). These OCR services use vocabulary dictionaries and the sequence of words in Persian sentences to increase detection accuracy, the features not utilized in our work. Therefore, we used separated words in our test samples to make the competition fair.

Table IV, Comparison with other works that use object detection for OCR. CRR is character recognition ratio and WRR is word recognition ratio.

| Ref. | Model | Description | CRR % |
|---|---|---|---|
| Year | Object type | | WRR% |
| [33] | YOLOv3 | Images of handwritten English text are processed. | 91.5 |
| 2022 | Words and Characters | | 70.8 |
| [34] | YOLOv4+ Spellcheck | Printed Arabic text in regular font is processed. | 96 |
| 2022 | Words and Characters | | 82 |
| [32] | Faster R-CNN | Balinese handwritten characters are processed. | 99.1 |
| 2023 | Characters | | - |
| [35] | YOLOv4 | Ottoman documents in regular font are processed. | 98.7 |
| 2024 | Characters | | - |
| This work | YOLOv5+ MLP | Words in two handwritten-style Persian fonts are processed. | 97.7 |
| | Words and Characters | | 95.3 |

To this end, we generated sample images with the same resolution and font size as our datasets, containing 308 four-lettered words (154 meaning words in Persian vocabulary and 154 words with no meaning in Persian vocabulary). By comparing the results for meaning and not meaning words, we can estimate the effect of using a dictionary. Generally, using a dictionary for OCR increases the detection accuracy of meaning words but decreases the detection accuracy for not meaning words. Samples are reproduced with Arial font (a very common font worldwide used for Persian and Arabic contents) as a control sample in addition to Maneli and IranNastaliq fonts. The results of this study are listed in Table V.

**Table V.** Comparison with OLOCR.com and Google’s Persian OCR services.

| | | Detection Accuracy (%) | | | | | |
|---|---|---|---|---|---|---|---|
| | | By Meaning Words | | By Meaningless Words | | By All Words | |
| Method | Font | CRR | WRR | CRR | WRR | CRR | WRR |
| **Google’s OCR** | **Maneli** | 94 | 87 | 60 | 15 | 77 | 51 |
| | **IranNastaliq** | 75 | 66 | 41 | 13 | 58 | 40 |
| | **Arial** | 98 | 98 | 91 | 73 | 95 | 85 |
| **OLOCR.com** | **Maneli** | 23 | 12 | 12 | 5 | 18 | 8 |
| | **IranNastaliq** | 17 | 10 | 7 | 0 | 11 | 5 |
| | **Arial** | 76 | 69 | 71 | 48 | 73 | 59 |
| **This Work** | **Maneli** | 98 | 98 | 98 | 97 | 98 | 98 |
| | **IranNastaliq** | 98 | 96 | 96 | 94 | 97 | 95 |
| | **Arial** | 100 | 100 | 100 | 100 | 100 | 100 |

By comparing the results from Google and OLOCR.com, we observe that both services produce acceptable results with the common Arial font. However, Google demonstrates better performance than OLOCR.com for Persian words due to the weakness of OLOCR.com in detection of Persian letters “پ, ژ, چ, گ” which are not in Arabic alphabet. In addition, we see that OLOCR.com does not show acceptable results with Maneli and IranNastaliq fonts despite its good results with the Arial font. This discrepancy highlights the intrinsic difference between Maneli and IranNastaliq handwriting-style fonts compared to the other fonts commonly used in Persian OCR. On the other hand, Google shows very good results with Maneli and IranNastaliq, indicating that this OCR service has been trained well with these fonts or similar ones. According to the results, Maneli font is generally easier to detect compared to IranNastaliq.

When comparing the results obtained from the proposed method with those from Google, we can see the potential of using object detection for recognizing handwriting-style fonts. Particularly when we see a notable excellence in detection of meaningless words. However, it is still premature to claim that our method is superior to Google's method, as this study considers a limited number of fonts and does not include process speed in the comparison. Specially knowing that the accuracy of Google on images containing meaning sentences with Maneli font is more that 99.5%. At least it can be said that using object detection in OCR of cursive and handwriting style fonts is worth more investigations.

## 5. Conclusion

The main challenge for OCR of cursive scripts is separation of characters, especially when the characters spatially overlap and are written in various forms. In this regard, we proposed a character detection model based on the object detection concept of YOLO neural network. We addressed the challenges of cursive scripts by focusing on Persian script as a case study and performed our investigation using handwriting-style fonts such as Maneli and IranNastaliq to show the advantages and disadvantages of the proposed detection process.

Initially, we developed a code for generation of synthetic datasets compatible with the YOLO network. The code can generate word image samples with complex fonts of Persian script and even for other languages with simple modifications. In this project, we created three datasets of arbitrary three-letter words, first with plain character forms of Maneli font, second with plain and alternative character forms of Maneli font, and third a mixture of Maneli and IranNastaliq characters. The YOLO network successfully detected and localized the characters in all datasets.

We observed that with the first dataset, evaluation metrics of precision, recall, F1-score, mAP50, and mAP50-95 reached 97.6%, 96.9%, 97.2%, 98.7%, and 95.2%, respectively. In the second dataset, there was a slight increase in all evaluation metrics for character detection. It is shown that by adding more samples, the accuracy can be increased even if more object forms are added. Additionally, with the third dataset, introduction of a new font resulted in no more than 1% reduction in evaluation metrics, demonstrating the network's stability in detecting characters with different fonts. Less than three thousand samples of three-letter word images were adequate to train the pre-trained object detection neural network model for detecting the characters of two handwriting-style fonts.

Another interesting point in this article occurred when we added artificial disturbances to images of the first dataset, such as noise, blur, and skew, ranging from low to high intensity, and the YOLO network performed well under various distortion conditions, even in specific conditions where the letters were hardly discernible to the human eye. We also combined these disturbances and observed that the YOLO network can accurately detect letters with multiple distortion conditions.

In the next step, we demonstrated that arrangement of the letters detected by YOLO to guess the intended word is easily applicable using a simple MLP neural network with the accuracy of 99.8% when trained and tested with Maneli font, and 97.7%, when trained and tested with a mixture of Maneli and IranNastaliq fonts, by using the localization information of the identified characters from YOLO as features and not the name of the font.

The presented method deals with the final stages of the OCR chain, which includes character recognition and word/text formation. Therefore, the OCR process loop is not yet complete. More efforts are still required to optimize the model for more characters, more fonts and even for other languages. Many other cases still need to be investigated with the proposed method in the real environment such as nonlinear image distortions, presence of background behind the text, and presence of shadows.

Investigation of handwritten-style fonts helps other researchers to improve OCR for these fonts or may be used to develop pre-trained networks for recognizing handwriting scripts using transfer learning, without the need for a large amount of handwritten data.

## 6. Reference

[1] M. Pandey, M. Arora, S. Arora, Ch. Goyal, V. K. Gera, and H. Yadav, “AI-based Integrated Approach for the Development of Intelligent Document Management System (IDMS)”, *Procedia Computer Science*, vol. 230, pp. 725-736, 2023.

[2] N. Girdhar, M. Coustaty, A. Doucet, “Digitizing History: Transitioning Historical Paper Documents to Digital Content for Information Retrieval and Mining—A Comprehensive Survey”, IEEE Transactions on Computational Social Systems, pp. 1-30, 2024.

[3] H.A. Alhamad, M. Shehab, M. K. Y. Shambour, M. A. Abu-Hashem, A. Abuthawabeh, H. Al-Aqrabi, M. Sh. Daoud, F. B. Shannaq, “Handwritten Recognition Techniques: A Comprehensive Review”, Symmetry, vol. 16, no. 6, p. 681, 2024.

[4] P. Shivakumara, U. Pal, “Cognitively Inspired Video Text Processing”, Springer Singapore, 2021.

[5] Z. Shen, R. Zhang, M. Dell, B. Charles, G. Lee, J. Carlson, W. Li, “Layoutparser: A unified toolkit for deep learning based document image analysis” In 16th International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland, September 5–10, pp. 131-146, 2021.

[6] J. Memon, M. Sami, R. A. Khan, M. Uddin, “Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)”, *IEEE access*, vol. 8, pp. 142642-142668, 2020.

[7] J. Park, E. Lee, Y. Kim, I. Kang, H.I. Koo, N.I. Cho, “Multi-lingual optical character recognition system using the reinforcement learning of character segmenter”, *IEEE Access*, vol. 8, pp. 174437-174448, 2020.

[8] Z. Khosrobeigi, H. Veisi, E. Hoseinzade, H. Shabanian, “Persian optical character recognition using deep bidirectional long short-term memory”, *Applied Sciences*, vol. 12, no. 22, p. 11760, 2022.

[9] M. Bonyani, S. Jahangard, M. Daneshmand, “Persian handwritten digit, character and word recognition using deep learning”, *International Journal on document analysis and recognition (IJDAR)*, vol. 24, no. 1, pp. 133-143, 2021.

[10] S. Ahmadi, M. Agarwal, A. Anastasopoulos, “PALI: A Language Identification Benchmark for Perso-Arabic Scripts”, In Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial). 2023.

[11] R. Azmi, E. Kabir, “A new segmentation technique for omnifont Farsi text”, *Pattern Recognition Letters*, vol. 22, no. 2, pp. 97-104, 2001.

[12] H. Khosravi, E. Kabir, “A blackboard approach towards integrated Farsi OCR system”, *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 12, pp. 21-32, 2009.

[13] V. Hajihashemi, M. M. A. Ameri, A. A. Gharahbagh, A. Bastanfard, “A pattern recognition based Holographic Graph Neuron for Persian alphabet recognition”, In 2020 Int. conf. on machine vision and image processing (MVIP), pp. 1-6. IEEE, 2020.

[14] V. Ghods, M.K. Sohrabi, “Online Farsi Handwritten Character Recognition Using Hidden Markov Model”, *Journal of Computers*, vol. 11, no. 2, pp. 169-175, 2016.

[15] J. Sadri, M.R. Yeganehzad, J. Saghi, “A novel comprehensive database for offline Persian handwriting recognition”, *Pattern Recognition*, vol. 60, p. 378, 2016.

[16] S. Khorashadizadeh, A. Latif, “Arabic/Farsi Handwritten Digit Recognition usin Histogra of Oriented Gradient and Chain Code Histogram”, *Int. Arab Journal of Information Technology (IAJIT)*, vol. 13, no. 4, 2016.

[17] M.J. Parseh, M. Meftahi, “A new combined feature extraction method for Persian handwritten digit recognition”, *International Journal of Image and Graphics*, vol. 17, no. 2, p. 1750012, 2017.

[18] G. A. Montazer, H. Q. Saremi, V. Khatibi, “A neuro-fuzzy inference engine for Farsi numeral characters recognition”, *Expert Systems with Applications*, vol. 37, no. 9, pp. 6327-6337, 2010.

[19] M. Pourreza, R. Derakhshan, H. Fayyazi, M. Sabokrou, “Sub-word based Persian OCR using auto-encoder features and cascade classifier”, In 2018 9th International Symposium on Telecommunications (IST), pp. 481-485. IEEE, 2018.

[20] Z.A. Aghbari, S. Brook, “HAH manuscripts: A holistic paradigm for classifying and retrieving historical Arabic handwritten documents”, *Expert Systems with Applications*, vol. 36, no. 8, pp. 10942-10951, 2009.

[21] Y. A. Nanehkaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, N. Al-Nabhan, “Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits”, *The Journal of Supercomputing*, vol. 77, pp. 3193-3222, 2021.

[22] M. Parseh, M. Rahmanimanesh, P. Keshavarzi, “Persian handwritten digit recognition using combination of convolutional neural network and support vector machine methods”, *The International Arab Journal of Information Technology*, vol.17, no. 4, pp. 572-578, 2020.

[23] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, H. Yu, “Deep learning for image inpainting: A survey”, *Pattern Recognition*, vol. 134, pp. 109046, 2023.

[24] S. Zhang, X. Lu, Z. Lu, “Improved CNN-based CatBoost model for license plate remote sensing image classification”, *Signal Processing*, vol. 213, p. 109196, 2023.

[25] S. Khosravi, A. Chalechale, “Chimp optimization algorithm to optimize a convolutional neural network for recognizing Persian/Arabic handwritten words”, *Mathematical Problems in Engineering*, vol. 1, p. 4894922, 2022.

[26] U. Hengaju, B. K. Bal, “Improving the Recognition Accuracy of Tesseract-OCR Engine on Nepali Text Images via Preprocessing”, *Advancement in Image Processing and Pattern Recognition*, vol. 3, no. 2, 3, pp. 40-52, 2023.

[27] M. M. Misgar, F. Mushtaq, S. S. Khurana, M. Kumar, “Recognition of offline handwritten Urdu characters using RNN and LSTM models”, *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 2053-2076, 2023.

[28] A. Mars, K. Dabbabi, S. Zrigui, M. Zrigui, “Combination of DE-GAN with CNN-LSTM for Arabic OCR on Images with Colorful Backgrounds”, In International Conference on Computational Collective Intelligence, pp. 585-596. Cham: Springer Nature Switzerland, 2023.

[29] M. F. Y. Ghadikolaie, E. Kabir, F. Razzazi, “Sub-word based offline handwritten farsi word recognition using recurrent neural network”, *ETRI Journal*, vol. 38, no. 4, pp. 703-713, 2016.

[30] R. Najam, S. Faizullah, “Analysis of recent deep learning techniques for Arabic handwritten-text OCR and Post-OCR correction”, *Applied Sciences*, vol. 13, no. 13, p. 7568, 2023.

[31] N. Ghanmi, A. Belhakimi, A. Awal, “CNN-BLSTM Model for Arabic Text Recognition in Unconstrained Captured Identity Documents”, In International Conference on Image Analysis and Processing, pp. 106-118. Cham: Springer Nature Switzerland, 2023.

[32] A. A. Pratama, M. D. Sulistiyo, A. F. Ihsan, “Balinese Script Handwriting Recognition Using Faster R-CNN”, *Journal of RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1268-1275, 2023.

[33] R. Mondal, S. Malakar, B. E.H. Smith, R. Sarkar, “Handwritten English word recognition using a deep learning based object detection architecture”, *Multimedia Tools and Applications*, vol. 81, pp. 975–1000, 2022.

[34] S. Alghyaline, “A Printed Arabic Optical Character Recognition System using Deep Learning”, *Journal of Computer Science*, vol. 18, no. 11, pp. 1038-1050, 2022.

[35] A. A. Demir, U. Ozkaya, “Ottoman character recognition on printed documents using deep learning”, Mühendislik Bilimleri ve Tasarım Dergisi, vol. 12, no. 2, pp. 392-402, 2024.

[36] X. Wang, S. Zheng, C. Zhang, R. Li, L. Gui, “R-YOLO: A real-time text detector for natural scenes with arbitrary rotation”, *Sensors*, vol. 21, no. 3, p. 888, 2021.

[37] D. Etter, S. Rawls, C. Carpenter, G. Sell, “A synthetic recipe for OCR”, In 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 864-869. IEEE, 2019.

[38] S. Hatami, S. Behnam, R. Shamsaee, “Improving detection of capsule endoscopy using YOLO”, *Tabriz journal of electrical engineering*, 2024, (In Persian), doi: 10.22034/tjee.2024.58239.4711.

[39] E. Zafarani-Moattar, M. R. Feizi-Derakhshi, A. Roohany, “The intelligent and automatic detection of type errors in large databases without using dictionary”, *Tabriz journal of electrical engineering*, vol. 47, no. 1, pp. 81-91, 2017, (In Persian)