Accepted manuscript (author version)

To appear in: **Majlesi Journal of Electrical Engineering (MJEE)**

Online ISSN: 2345-377X **Print ISSN:** 2345-3796

This PDF file is not the final version of the record. This version will undergo further copyediting, typesetting, and production review before being published in its definitive form. We are sharing this version to provide early access to the article. Please be aware that errors that could impact the content may be identified during the production process, and all legal disclaimers applicable to the journal remain valid.

Received: 03-Jun-2025	**Revised:** 24-Jun-2025
Accepted: 12-Jul-2025	
DOI: 10.57647/j.mjee.2025.17063	

Original Research

Towards Omni-Font Optical Character Recognition (OCR) for Persian script using the YOLO object detection model

Mojtaba Gandomkar, Sahar Khoramipour

Mojtaba Gandomkar
Department of Electrical and Computer Engineering, Jundi-Shapur University of Technology, Dezful, Iran.
Email: gandomkar@jsu.ac.ir
Corresponding author
ORCID: 0000-0003-0832-7799

Sahar Khoramipour
Department of Electrical and Computer Engineering, Jundi-Shapur University of Technology, Dezful, Iran.
Email: saharkhoramipour17@gmail.com

© Author(s) 2025

Accepted Manuscript: Author Version

Abstract

Optical Character Recognition (OCR), especially for scripts with complex structures like Persian script, faces significant challenges in interpreting nuanced characters and contextual variations. This study provides a straightforward and scalable approach to developing omni-font OCR systems. A synthetic dataset incorporating words, numbers, punctuation marks, mathematic symbols, and whitespace characters is developed to evaluate YOLO's capability in detecting 70 characters across 15 formal, informal, and handwritten-style fonts. The proposed method for detection of regular space and non-breaking space characters achieved high-

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

precision results that may eliminate the need for a separate word detection stage in an OCR system. In another investigation, we attempted to detect characters from an unseen font by training the model on a batch of other fonts. A formal font such as “B Nazanin” is near-completely detectable without being directly included in the model’s training with a batch of fourteen other fonts. For a handwritten-style font such as “MRT_Sayeh-1”, the mean detectability increases from 54%, when training the model with other single fonts, to 80% when a batch of fourteen other fonts is used. Overall, this study demonstrates that object detection-based OCR models have the potential for Omni-Font text recognition through expanded datasets and advancements in deep learning.

KEYWORDS:

Optical Character Recognition, Object Detection, YOLO, Persian Script, Handwritten-style fonts.

Accepted Manuscript: Author Version

1. INTRODUCTION

Optical Character Recognition (OCR) systems convert scanned or digital text into machine-readable formats through image digitization, text segmentation, and character recognition. By transforming image-based documents into searchable and editable text, OCR reduces manual workload and improves efficiency. These systems play a crucial role in automating tasks such as license plate recognition [1], [2], bank check processing [3], mail sorting [4], and signature verification [5]. Preprocessing techniques, such as noise reduction and skew correction, are applied to improve the accuracy. Subsequently, segmented words and characters undergo feature extraction for classification. The final output may include error correction mechanisms to enhance recognition accuracy [6]. Recent advancements in OCR systems have significantly improved the recognition of both printed and handwriting text [7]. However, considerable challenges remain, requiring further research and innovative solutions.

Despite technological advancements in OCR for Latin-based scripts, fewer studies are available on non-Latin scripts such as Persian, Arabic, and Urdu [8]. The complex structure and cursive nature of these languages pose significant challenges for accurate text recognition.

Despite their grammatical differences, Persian and Arabic share many writing features. Both languages use a right to left script for letters, while embedded numerals and symbols are written left to right. Additionally, Persian includes four extra letters (“ژ“ ,”پ“ ,”چ“ ,”گ”) that are not found in Arabic. The Persian alphabet consists of 32 letters, which have context-dependent forms and no capital forms. Some letters appear in only two forms (final and isolated), while others have four distinct forms (initial, medial, final, and isolated). Many letters have similar structures, distinguished primarily by dot placement. Table 1 presents the context-dependent forms of Persian letters, numbers, mathematical symbols, punctuation marks, and two whitespaces characters examined in this study, along with their corresponding class numbers.

Advanced analysis algorithms have been developed to improve the OCR efficiency of Persian script, addressing complexities such as letter connectivity, bidirectional text flow, positional shape variations, and font diversity [9], [10], [11]. Many traditional methods for feature extraction and classification have been explored in the literature, including Hidden Markov Models (HMM) [12], Principal Component Analysis (PCA) [13], Histogram of Oriented Gradients (HOG) [14], Scale-Invariant Feature Transform (SIFT) [15], Directional Chain Code (CCH) [16], Wavelet Transform [17], Gabor filtering [18], and Clustering [19] for feature extraction, as well as Support Vector Machine (SVM) [8], [20], [21], Multi-Layer Perceptron (MLP) [22], Decision Tree [23], and k-Nearest Neighbors (KNN) [19] for classification. Although traditional techniques improve accuracy, they require extensive feature tuning and preprocessing, which can increase computational complexity and reduce efficiency.

Table 1. Letter forms and characters in Persian script along with their corresponding class numbers.

Class number	Character name	Independent	Initial	Medial	Final	Class number	Character name	Independent	Initial	Medial	Final
0	Aaa	آ	-	-	ـآ	21	Ein	ع	عـ	ـعـ	ـع
1	Alf	ا	-	-	ـا	22	Ghein	غ	غـ	ـغـ	ـغ

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

2	Beh	ب	ﺑ	ﺒ	ﺐ
3	Peh	پ	ﭘ	ﭙ	ﭗ
4	Teh	ت	ﺗ	ﺘ	ﺖ
5	Seh	ث	ﺛ	ﺜ	ﺚ
6	Jim	ج	ﺟ	ﺠ	ﺞ
7	Che	چ	ﭼ	ﭽ	ﭻ
8	Hej	ح	ﺣ	ﺤ	ﺢ
9	Khe	خ	ﺧ	ﺨ	ﺦ
10	Dal	د	-	-	ﺪ
11	Zal	ذ	-	-	ﺬ
12	Reh	ر	-	-	ﺮ
13	Zeh	ز	-	-	ﺰ
14	Zhe	ژ	-	-	ﮋ
15	Sin	س	ﺳ	ﺴ	ﺲ
16	Shin	ش	ﺷ	ﺸ	ﺶ
17	Sad	ص	ﺻ	ﺼ	ﺺ
18	Zad	ض	ﺿ	ﻀ	ﺾ
19	Taa	ط	ﻃ	ﻄ	ﻂ
20	Zaa	ظ	ﻇ	ﻈ	ﻆ
23	Feh	ف	ﻓ	ﻔ	ﻒ
24	Ghaf	ق	ﻗ	ﻘ	ﻖ
25	Kaf	ک	ﮐ	ﮑ	ﮏ
26	Gaf	گ	ﮔ	ﮕ	ﮓ
27	Lam	ل	ﻟ	ﻠ	ﻞ
28	Mim	م	ﻣ	ﻤ	ﻢ
29	Nun	ن	ﻧ	ﻨ	ﻦ
30	Vav	و	-	-	ﻮ
31	Heh	ه	ﻫ	ﻬ	ﻪ
32	Yeh	ی	ﯾ	ﯿ	ﯽ
33	Hmz	ء	-	-	-
34	Ahmz	أ	-	-	ﺄ
35	Vhmz	ؤ	-	-	ﺆ
36	Ehmz	ئ	ﺋ	ﺌ	ﺊ
37 to 46		1234567890			
47 to 58		[(».،:؛!؟«)]			
59 to 66		<=+-/×÷>			
67	MMYZ	Decimal sign			
68	HSPC	Non-breaking space			
69	SPC	Regular space			

Recent advancements in deep learning have significantly improved Persian and Arabic character recognition, enhancing both accuracy and efficiency. Convolutional Neural Networks (CNNs) have optimized the feature extraction and classification by capturing complex patterns in raw data. Elleuch et al. (2016) employed CNNs for Arabic character feature extraction, followed by SVM classification [24]. Ashraf et al. (2017) introduced a simple CNN (SCNN) inspired by LeNet-5, as well as an ensemble CNN (ECNN) with optimized parameters for Persian OCR [25]. Mohammed Aarif et al. (2020) employed deep CNNs and transfer learning for handwritten Urdu OCR, achieving superior accuracy with OCR-AlexNet and OCR-GoogleNet compared to traditional methods [26]. Bonyani et al. (2021) evaluated CNN-based architectures, including DenseNet, ResNet, and VGG, for Persian OCR and identified DenseNet as the most effective model for complex Persian texts [13]. Khosravi et al. (2022) optimized the LeNet-5 CNN for Persian and Arabic handwritten character recognition using the Chimpanzee Optimization Algorithm (ChOA) [27].

Recent researches highlight the effectiveness of integrating Long Short-Term Memory (LSTM) networks with other models for character recognition. Naseer et al. (2019) combined LSTM-RNN with CNN for Urdu OCR [28]. Similarly, Khosrobiegi et al. (2022) employed a deep Bidirectional LSTM (BLSTM) alongside CNN to enhance Persian OCR [8].

Text recognition in historical books and handwritten scripts is challenging due to low print quality, paper degradation, unknown fonts, handwriting variations, and overlapping characters. Soheili et al. (2017) addressed these challenges by integrating subword clustering with an LSTM network [29]. Similarly, Mousavi et al. (2017) developed a method for extracting text from Achaemenid inscriptions using Tesseract 3.04 [30].

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm that rapidly identifies and localizes multiple objects in images in real time. Mondal et al. (2022) introduced a lexicon-free handwriting recognition method using YOLOv3, enabling sequential character detection without relying on large datasets or predefined dictionaries [31]. Similarly, Demir et al. (2022) applied a YOLO-based model for character recognition in Ottoman documents, demonstrating strong performance in historical text recognition [32].

In Persian OCR, YOLO is widely applied to tasks such as license plate recognition [33], ID card detection [34], and text localization [35]. However, its use for character-level recognition within Persian words has been less explored. This study investigates the application of the YOLO model for Persian character recognition, aiming to assess its capability in establishing an omni-fonts OCR system.

In Persian text recognition, a key challenge is the lack of large, diverse datasets, often resulting in the creation of custom, limited datasets for model training. Character recognition with object detection models such as YOLO, which require their own specific datasets, makes finding a suitable dataset even more challenging. In our previous work, we developed a code capable of generation an unlimited number of Persian words images, suitable for training the YOLO model for character-level recognition of Persian script [36]. We applied this approach to Maneli and IranNastaliq-web handwritten-style fonts. We demonstrated that character detection using this method does not require a large number of training samples even for handwritten-style fonts. Additionally, we showed that this approach is robust against significant noise, skewness, and blurriness in image samples. In this study, we extended the code to include more characters, such as numbers, punctuation marks, mathematical symbols, and whitespace characters, enabling the generation of unlimited word variations with 70 distinct characters.

The detection of whitespaces, including regular spaces and non-breaking spaces, is a key innovation of this study. Non-breaking spaces are crucial in Persian orthography, yet their detection within words has received limited attention due to the challenges faced by traditional methods. Recognizing regular spaces is also vital for

Accepted Manuscript: Author Version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

accurate word segmentation. While existing approaches focus on word separation before OCR processing, the findings of this study suggest that this step may not be necessary in OCR systems. To localize a whitespace character, we propose using bounding boxes that encompass the preceding and following characters. The results of this simple yet effective approach were highly promising, demonstrating the power of the YOLO object detection model in addressing this challenge.

To further enhance the diversity of the dataset and evaluate character recognition across different typographic styles, 15 frequently used fonts were carefully selected from various styles, as illustrated in Figure 1, allowing for a more comprehensive analysis of character recognition. The main focus is to assess whether a system can be trained to recognize characters across multiple fonts. Specifically, we will examine whether a model trained on a batch of fonts can recognize characters from a font that was not included in the training process. In this case, increasing the number of fonts in training may enable the development of a system capable of recognizing almost all fonts with sufficient accuracy. Similarly, training on the handwriting of different individuals could facilitate the creation of a natural handwriting OCR system.

Some of the selected fonts are optimized for readability and formal content presentation, ensuring clarity under various reading conditions. Others are commonly used in informal text or visually engaging designs, enhancing expressive communication. The remaining fonts were chosen for their close resemblance to natural handwriting, making them particularly valuable for future research in handwritten optical character recognition (HOCR) using object detection models. The font selection process, given the wide range of available fonts, involved a degree of subjectivity, as establishing strict boundaries between font characteristics is challenging.

Font Name	Text Sample
BNazanin (B Nazanin)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BZar (B Zar)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BMitra (B Mitra)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BLotus (B Lotus)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BTraffic (B Traffic)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BYekan (B Yekan)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
Tahoma	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
Arial	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BKamran (B Kamran)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BMehr (B Mehr)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
BTabassom (B Tabassom)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
INastaliq (IranNastaliq-Web)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
Maneli	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
DastNevis (Dast Nevis)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.
Sayeh1 (MRT_Sayeh-1)	عجب! از پنجره‌ی باز مدرسه، صدای گرم خنده و شادی می‌آید.

Fig. 1. An example of text in 15 frequently used fonts, ranging from formal to handwritten-style fonts.

2. METHODS

2.1. Dataset Generation

In this study, we generate a synthetic dataset to demonstrate the capability of the YOLO object detection model for OCR of Persian scripts, aiming for character-level recognition in omni-font printed documents. The image generation process involves writing words in black text on a white background, then converting each letter into a distinct color while preserving its form and position [36]. This method ensures precise spatial positioning of characters, even when they overlap, aiding YOLO in accurately identifying each character's bounding box.

The dataset is generated with minimal user intervention. The user only provides a set of words and specifies the font name. This approach is adaptable to other fonts and languages with minor modifications. This concise and automated dataset generation process is key to demonstrating the idea of using object detection models for Persian OCR, enabling efficient character-level recognition across diverse fonts.

We expand the dataset to include 15 frequently used Persian fonts, allowing for a more comprehensive analysis compared to our previous work, which focused on two handwritten-style fonts (Maneli and IranNastaliq-web).

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

The selected fonts encompass a wide range of typographic styles, from formal and highly readable fonts to those that closely resemble handwritten scripts. This diversity presents a challenge for the YOLO model in distinguishing between different forms of the same character.

In this study, we utilized a dataset comprising 70 primary Persian characters, including 32 letters of the Persian alphabet, selected Arabic characters “آ, ؤ, ئ, أ, ء”, numerals with decimal point, basic mathematical symbols “<, =, +, -, ×, ÷, >”, punctuation marks “[, (, », ., ،, :, ؟, !, ؛, «,),]”, and two whitespace characters, including regular space and non-breaking space (Table 1).

These characters were selected based on Persian orthographic standards and the standard Persian keyboard layout to ensure comprehensive coverage. However, certain symbols, such as “%” and “{}”, were excluded due to font rendering inconsistencies in some fonts. Additionally, symbols like “@, #, ~”, elongation marks (Kashida or Tatweel), and Arabic diacritics (fatḥah, kasrah, ḍammah, and tanwīns) were omitted for simplicity, as they are rarely used in public Persian texts. Different forms of a letter within a word (initial, medial, final, and independent), as well as its variations when combined with other characters, are all labeled with the original letter name. This approach simplifies the process, leveraging the YOLO model’s capability to detect various forms of small objects.

Whitespace characters play a crucial role in enhancing the readability of Persian script. Regular space ensures clear separation between words, while non-breaking spaces maintain the integrity of word components. In this study, to localize a whitespace character, the corresponding bounding box is designed to be drawn in such a way that it encompasses one character before and one character after the whitespace, as illustrated in Figure 2. This dynamic bounding strategy enables the model to handle kerning variations and irregular spacing across a wide range of fonts, including complex handwritten styles such as INastaliq and Maneli.

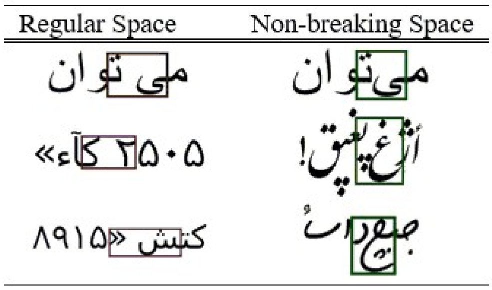

Fig. 2. Bounding boxes for regular space and non-breaking space characters, as proposed for whitespace detection.

Figure 3 presents image samples of character strings in different fonts along with their corresponding bounding boxes. To ensure a balanced and comprehensive dataset for training, validation, and testing, we generated 1,800 images per font. Among them, 1,000 images are allocated for training, 200 images for validation and model optimization during training, and 600 images for final testing. Each image contains a structured combination of characters to reflect real-world Persian text variations, including:

- A three-letter word followed by a whitespace, or mathematical operator, then a four-letter word.
- A four-letter word followed by a whitespace, or mathematical operator, then a three-letter word.
- A three-letter word followed by a regular space, or mathematical operator, then a four-digit decimal number.
- A four-digit decimal number followed by a regular space, or mathematical operator, then a three-letter word.

Additionally, punctuation marks such as periods, commas, and parentheses have been incorporated where appropriate to enhance the dataset’s realism and complexity. This structured approach ensures that the model is exposed to diverse typographic and syntactic variations, strengthening its ability to recognize Persian characters across different contexts. The words do not necessarily have meaning in Persian language.

During dataset generation, the frequency of each character’s appearance was carefully adjusted based on its structural diversity, relative size, and visual similarity to other characters. For instance, early experiments revealed challenges in distinguishing certain paired symbols, such as “(), [], <>, and «»” primarily due to their symmetry and resemblance. To address this issue, the occurrence of these symbols was strategically increased in the dataset, enabling the model to learn their distinct features more effectively and improving overall recognition accuracy.

Accepted Manuscript Author Version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

Fig. 3. Samples of the dataset with different fonts indicating their corresponding bounding boxes.

To investigate the feasibility of creating an omni-font OCR system, we examine how accurately the model can detect a new font that was not included in the training process. To this end, we reinforced the YOLO model by training it not only on individual fonts but also on batches of different fonts. In practice, this was achieved by combining the datasets of the fonts in each batch. The fonts included in each batch are shown in Figure 4. These batches consist of five batches of three fonts, labeled “A” to “E”, and three batches of five fonts, labeled “F”, “G”, and “H”. Additionally, for each font, we created a batch of fourteen fonts by excluding that specific font from the list. Efforts were made to include a diverse range of formal, informal, and handwritten-style fonts in each batch. Among various possible arrangements, this setup appears sufficient for the purpose of this article.

Font Batch Labels		Fonts in the batch
3-font batches	A	BNazanin, BYekan, BTabassom
	B	BZar, Tahoma, INastaliq
	C	BMitra, Arial, Maneli
	D	BLotus, BKamran, DastNevis
	E	BTraffic, BMehr, Sayeh1
5 -font batche	F	BNazanin, BLotus, Tahoma, BMehr, Maneli
	G	BZar, BTraffic, Arial, BTabassom, DastNevis
	H	BMitra, BYekan, BKamran, INastaliq, Sayeh1

Fig. 4. Font batch labels and the fonts included in each of them.

2.2. YOLO algorithm

YOLO, initially introduced in 2016, quickly became a leading object detection framework, outperforming traditional methods in both speed and accuracy [37]. As a deep convolutional neural network, YOLO combines feature extraction through convolutional layers with bounding box prediction via fully connected layers, enabling efficient real-time object detection. Developed within the DarkNet framework, it provides a unified approach for simultaneous object classification and localization [38], [39].

In this study, we leverage YOLO’s object detection capabilities to develop an OCR system tailored for cursive scripts, using Persian as a case study. By formulating character recognition as an object detection task, our approach enables precise localization and classification of characters across diverse font styles, significantly enhancing OCR performance in complex script structures. Over the years, multiple improved versions of YOLO have been introduced, further enhancing its performance and adaptability.

The training process was conducted using YOLO11n, which achieved an optimal balance between accuracy and computational efficiency. Although larger versions, such as YOLO11s and YOLO11m, were also evaluated, they did not yield substantial improvements relative to their increased training time in this study. However, for more complex tasks, such as recognizing a wider range of fonts or natural human handwriting, larger or customized YOLO models may be required.

During the training process, we employed the default hyperparameters of YOLO11n, along with its standard augmentation and mosaic settings available at the time of this study. However, given the presence of symmetrical characters in Persian script, horizontal reflections were disabled during mosaicking, contrary to YOLO's defaults. Vertical reflections, which are not applied by default in YOLO, were also omitted. Following YOLO’s standard training procedure, mosaic augmentation was disabled for the final 10 epochs. A batch size of 16 was selected, and training continued until no further performance improvements were observed. After each epoch, the model is

Accepted Manuscript author version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

evaluated and the final trained model is the one with the best performance during training process.

To evaluate the impact of input image resolution, we conducted experiments by reducing the default size of 640×640 pixels to 480×480 pixels. This reduction led to a noticeable decline in performance. Consequently, the default image size was retained to preserve accuracy. Fine-tuning training parameters for real-world applications remains an open area for further research, which is beyond the scope of this study.

2.3. Evaluation criteria

Model performance is evaluated using standard metrics, including precision, recall, F1-score, and mean Average Precision (mAP). The formulas for these evaluation metrics are provided below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, \quad (2)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (3)$$

$$\text{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} AP_{\text{k}} \quad (4)$$

Here, True Positive (TP) represents the number of correctly identified positive instances, while False Positive (FP) refers to cases where an object is incorrectly classified as positive. True Negative (TN) denotes the number of correctly identified negative instances, whereas False Negative (FN) corresponds to cases where positive instances are misclassified as negative.

Mean Average Precision (mAP) is a widely used metric in object detection and classification tasks, measuring the average precision across all classes in the dataset. Here, n represents the total number of classes, and AP_{k} denotes the average precision of class k.

To further analyze model performance, we introduce additional parameters that quantify the detectability and perceptivity of fonts relative to each other. We define the perceptivity of font i on font j (P_{ij}) as a measure of how accurately a model trained on font i can recognize characters from font j. Similarly, the detectability of font i by font j or font batch j (D_{ij}) is defined as the precision of character recognition for font i, when a YOLO model trained on font j or font batch j is used. Clearly, for two individual fonts, $P_{ij} = D_{ji}$. Additionally, we expect $D_{ji} \approx D_{ij}$ for two individual fonts, indicating a closely symmetric relationship in recognition capabilities.

Additionally, we define two aggregate measures: the mean perceptivity of font i (mP_i), which quantifies the average detectability of other fonts by a model trained on font i, and the mean detectability of font i (mD_i), which represents the average detectability of font i using models trained on batches of other fonts, excluding the target font itself.

$$\text{mP}_i = \frac{1}{n-1}\sum_{j=1}^{j=n} \text{P}_{ij}, \ j \neq i \quad (5)$$

$$\text{mD}_i = \frac{1}{k}\sum_{j=1}^{j=k} \text{D}_{ij} \quad (6)$$

where n = 15 (number of fonts), and k is number of batches of fonts that do not include font i. These metrics offer insight into the generalizability of character recognition across diverse typographic styles and fonts.

3. RESULTS

To initiate our analysis, we examine the correlation between the number of training samples and model performance using BNazanin and Sayeh1 fonts as case studies. As illustrated in Figure 5, the evaluation metrics exhibit a rapid improvement as the number of training samples increases. Expectedly, a formal and highly readable font like BNazanin requires fewer samples compared to a handwritten-style font such as Sayeh1. With only 250 samples, all performance indicators exceed 90% for both fonts. Accuracy reaches a satisfactory level after 300 samples, and the metrics gradually stabilize, approaching their asymptotic values beyond approximately 700 samples.

Accepted Manuscript: Author Version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

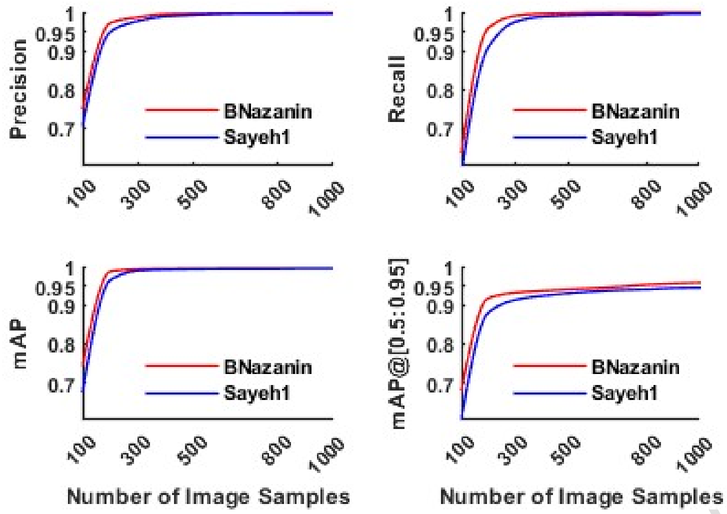

Fig. 5. Increment of evaluation metrics by increasing the number of samples used for training.

Figure 6 illustrates the confusion matrix for training the YOLO model with Sayeh1 font as an example, which is one of the handwritten-style fonts. The model was trained on 1,000 samples and evaluated on 600 test samples. The results demonstrate that the model effectively detects all characters with minimal misclassification. It successfully distinguishes visually characters (e.g., those with different dot placements) without confusion. Notably, the model exhibits robust performance in recognizing small and symmetrical characters. In addition, it accurately localizes whitespaces characters. These findings underscore the model's resilience in handling diverse and complex character structures, highlighting the effectiveness of the proposed method in whitespace detection.

Accepted Manuscript: Author Version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

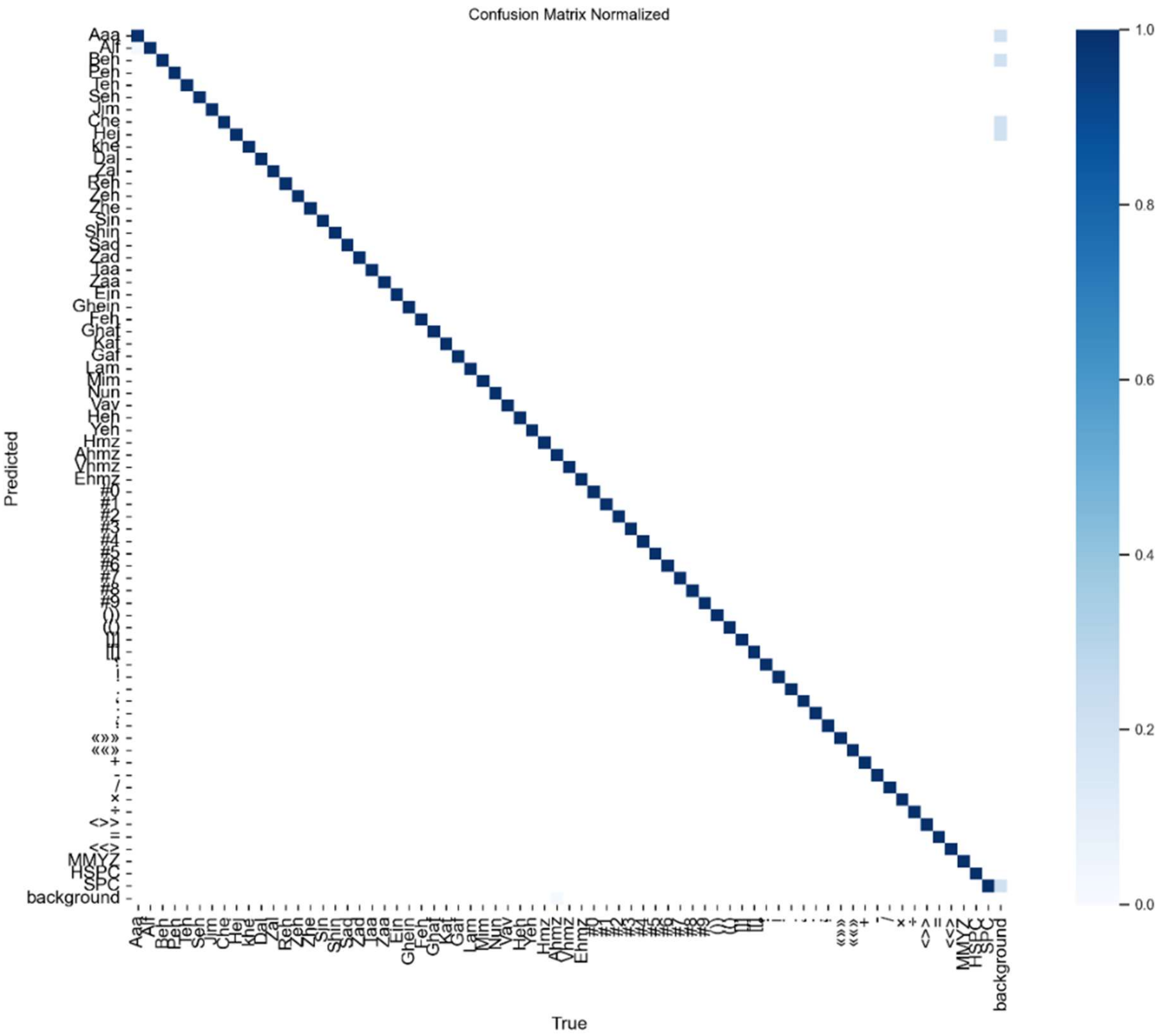

Fig. 6. Confusion Matrix for the model trained with 1000 samples and tested on 600 samples of Sayeh1 font.

Figure 7 illustrates the detectability of each font by another font used for training the YOLO model. For instance, the first row represents the detectability of all fonts when the model is trained on the BNazanin dataset. The detectability of BKamran by BNazanin is 67%, whereas, in reverse, BNazanin's detectability by BKamran is slightly higher at 73%. While the detectability scores are not perfectly symmetric (i.e., $D_{ij} \neq D_{ji}$), a notable relative symmetry is observed ($D_{ij} \approx D_{ji}$). This indicates that when a font is capable of recognizing another font well, it is likely that the reverse is true too, though the exact values may slightly differ.

The last row of Figure 7 presents the mean detectability (mD) for each font. Formal fonts, commonly used in academic texts, exhibit a mean detectability of approximately 80%, indicating a high degree of similarity among them. In contrast, handwritten-style fonts demonstrate a considerably lower mean detectability of around 50%, highlighting the unique and diverse visual characteristics of each handwritten style. These results suggest that formal fonts, due to their standardized typographic design, share greater resemblance with one another, whereas handwritten fonts are more distinct and individualized. This discrepancy underscores the challenge that OCR systems face when recognizing handwritten fonts, as they must accommodate a broader range of complex visual variations.

Figure 8 presents the mean perceptivity (mP) of each font relative to other fonts and the mean detectability (mD) of each font when recognized by other fonts. The fonts are arranged based on their mP values, and this ordering is consistently referenced throughout the paper. For most fonts, mP and mD values are closely aligned. However, in certain cases, such as INastaliq and BKamran, the mP is approximately 10% higher than the mD. This suggests that these fonts significantly contribute to detecting other fonts when used for training the model, whereas other fonts exhibit limited capability in recognizing them.

These observations highlight a key challenge in OCR for handwritten scripts, where the variance in character shapes is more pronounced compared to formal fonts. The higher mP and mD values observed in formal fonts suggest that OCR models tend to perform better in recognizing characters across different formal fonts due to

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

their standardized visual features. In contrast, the lower mP and mD values for handwritten fonts reflects their greater individuality and complexity, often leading to lower recognition accuracy. In summary, these results indicate that OCR systems may achieve more reliable performance with formal fonts, given their consistent design and visual regularity. However, improving recognition accuracy for handwritten fonts will likely require more advanced techniques that account for the increased variation in character shapes.

The font used for model training \ The font that the trained model is tested on	BNazanin	BZar	BMitra	BLotus	BTraffic	BYekan	Tahoma	Arial	BKamran	BMehr	BTabassom	INastaliq	Maneli	DastNevis	Sayeh1
BNazanin	1.00	0.97	0.98	0.95	0.90	0.78	0.83	0.94	0.67	0.77	0.79	0.58	0.56	0.57	0.53
BZar	0.99	1.00	0.97	0.98	0.85	0.70	0.77	0.92	0.73	0.74	0.72	0.62	0.62	0.53	0.58
BMitra	0.97	0.97	1.00	0.96	0.85	0.76	0.84	0.91	0.71	0.67	0.77	0.60	0.53	0.53	0.54
BLotus	0.96	0.99	0.95	1.00	0.87	0.72	0.75	0.90	0.72	0.71	0.73	0.59	0.62	0.49	0.55
BTraffic	0.90	0.86	0.85	0.85	1.00	0.82	0.83	0.81	0.71	0.81	0.70	0.51	0.61	0.54	0.52
BYekan	0.86	0.75	0.81	0.72	0.84	1.00	0.87	0.78	0.71	0.75	0.74	0.61	0.57	0.57	0.54
Tahoma	0.91	0.84	0.89	0.82	0.89	0.85	1.00	0.84	0.60	0.76	0.69	0.48	0.50	0.49	0.48
Arial	0.93	0.90	0.88	0.92	0.77	0.66	0.73	1.00	0.61	0.71	0.65	0.62	0.51	0.52	0.53
BKamran	0.73	0.73	0.77	0.77	0.73	0.68	0.61	0.62	1.00	0.67	0.76	0.57	0.59	0.64	0.61
BMehr	0.79	0.78	0.71	0.69	0.81	0.80	0.70	0.69	0.60	1.00	0.59	0.59	0.60	0.54	0.58
BTabassom	0.78	0.71	0.82	0.73	0.76	0.73	0.64	0.66	0.66	0.66	1.00	0.46	0.57	0.50	0.47
INastaliq	0.69	0.74	0.63	0.70	0.70	0.59	0.63	0.59	0.58	0.55	0.51	0.99	0.58	0.57	0.54
Maneli	0.62	0.58	0.52	0.56	0.58	0.69	0.61	0.53	0.50	0.67	0.52	0.53	1.00	0.60	0.61
DastNevis	0.61	0.61	0.56	0.62	0.51	0.57	0.52	0.48	0.53	0.62	0.54	0.42	0.53	0.99	0.56
Sayeh1	0.51	0.52	0.48	0.49	0.57	0.55	0.45	0.51	0.54	0.54	0.51	0.46	0.61	0.62	1.00
Mean Detectability	0.80	0.78	0.77	0.77	0.76	0.71	0.70	0.73	0.63	0.69	0.66	0.54	0.57	0.55	0.54

Fig. 7. Detectability of every font using a model trained with the font written on the left of row.

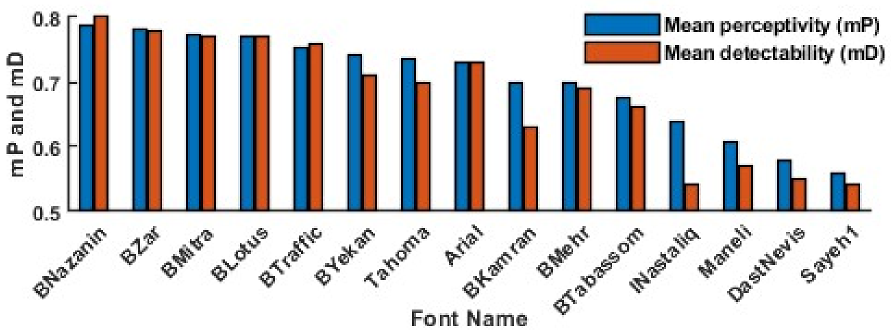

Fig. 8. Mean perceptivity (mP) of every font on the other fonts and mean detectability (mD) of every font by the other fonts.

Figures 9 and 10 illustrate the detectability (D) of each font when the model is trained using different combinations of other fonts. Specifically, Figure 9 presents the results obtained from training the model with batches of three fonts, while Figure 10 shows the outcomes when training is conducted with batches of five fonts. Furthermore, the mean detectability (mD) of each font is reported in the final row of these figures.

The results clearly indicate that expanding the range of fonts used during training significantly enhances character recognition performance across all fonts. Compared to Figure 8 or the last row of Figure 7, training the model with three different fonts leads to a noticeable improvement in mean detectability (mD) for all fonts, with top-performing formal fonts achieving mD values exceeding 95%. Specifically, training with three fonts increases mD from 12% for the BYekan font to 20% for BMitra and Arial fonts. Expanding training to five fonts, further boosts mD, surpassing 99% for formal fonts. Additionally, the mD for the handwritten-style font Sayeh1, which was 54% when trained with individual fonts, rises to 68% with batches of three fonts and further improves to 73% with batches of five fonts.

Figure 11 evaluates the mean detectability (mD) of each font when the model is trained using a batch of fourteen other fonts. Notably, formal fonts are recognized with near-perfect accuracy, even without direct training on those specific fonts. In contrast, the mD for the handwritten-style font Sayeh1 shows a substantial improvement, reaching 80% as more fonts are incorporated into training. These results underscore the importance

 This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

of including diverse font categories in the training process, as it enhances the model's ability to detect previously unseen fonts and improves its generalization for omni-font recognition.

Font batch for model training	BNazanin	BZar	BMitra	BLotus	BTraffic	BYekan	Tahoma	Arial	BKamran	BMehr	BTabassom	INastaliq	Maneli	DastNevis	Sayeh1
A	1.00	0.98	0.99	0.95	0.94	1.00	0.97	0.95	0.75	0.92	1.00	0.62	0.61	0.58	0.56
B	0.99	1.00	0.99	0.99	0.97	0.90	1.00	0.96	0.86	0.85	0.85	0.99	0.76	0.64	0.65
C	1.00	0.98	1.00	0.98	0.86	0.79	0.88	1.00	0.82	0.79	0.80	0.73	1.00	0.71	0.80
D	0.97	0.98	0.97	1.00	0.87	0.76	0.84	0.91	1.00	0.88	0.83	0.73	0.82	0.99	0.72
E	0.93	0.91	0.92	0.91	1.00	0.88	0.93	0.89	0.83	1.00	0.84	0.69	0.78	0.79	1.00
Mean Detectability	0.97	0.96	0.97	0.96	0.91	0.83	0.90	0.93	0.81	0.86	0.83	0.69	0.74	0.68	0.68

(Column header: The font that the trained model is tested on)

Fig. 9. Detectability and mean detectability of different fonts relative to five batches of fonts, each including three diverse fonts.

Font batch for model training	BNazanin	BZar	BMitra	BLotus	BTraffic	BYekan	Tahoma	Arial	BKamran	BMehr	BTabassom	INastaliq	Maneli	DastNevis	Sayeh1
F	1.00	1.00	0.99	1.00	0.96	0.92	1.00	0.97	0.86	1.00	0.87	0.75	1.00	0.78	0.75
G	1.00	1.00	0.99	0.98	1.00	0.94	0.93	1.00	0.89	0.96	1.00	0.76	0.79	0.99	0.72
H	1.00	0.99	1.00	0.98	0.97	1.00	0.96	0.97	1.00	0.90	0.91	0.99	0.83	0.80	1.00
Mean Detectability	1.00	0.99	0.99	0.98	0.96	0.93	0.94	0.97	0.87	0.93	0.89	0.75	0.81	0.78	0.73

(Column header: The font that the trained model is tested on)

Fig. 10. Detectability and mean detectability of different fonts relative to three batches of fonts, each including five diverse fonts.

	BNazanin	BZar	BMitra	BLotus	BTraffic	BYekan	Tahoma	Arial	BKamran	BMehr	BTabassom	INastaliq	Maneli	DastNevis	Sayeh1
Mean detectability of each font by the model trained with fourteen other fonts	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.95	0.97	0.92	0.84	0.85	0.87	0.80

Fig. 11. Mean detectability of different fonts using a batch of fourteen other fonts in training.

Figure 12 illustrates the mean detectability (mD) of each font when the model is trained with batches containing different numbers of other fonts. These batches include single-font, three-font, five-font, and fourteen-font training sets. The results demonstrate a clear trend: as the number of fonts included in the training increases, the overall efficiency of character recognition improves.

When the model is trained with fourteen fonts, formal fonts are recognized with near-perfect accuracy, even without direct training on those specific fonts. Interestingly, handwritten-style fonts also show substantial improvements under these conditions. The mD for Maneli and DastNevis fonts surpasses 85%, while for INastaliq and Sayeh1 fonts is over 80%.

While part of this increase in mD can be attributed to the inclusion of simpler elements, such as numbers and punctuation marks, the findings suggest that further improvements may be achieved by increasing the diversity of fonts during training. This observation highlights the potential of object detection-based OCR systems in recognizing even highly variable handwriting styles. By constructing a sufficiently large and diverse training dataset, it may be possible to achieve highly accurate recognition of natural handwriting, further narrowing the performance gap between OCR systems for printed and handwritten scripts.

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

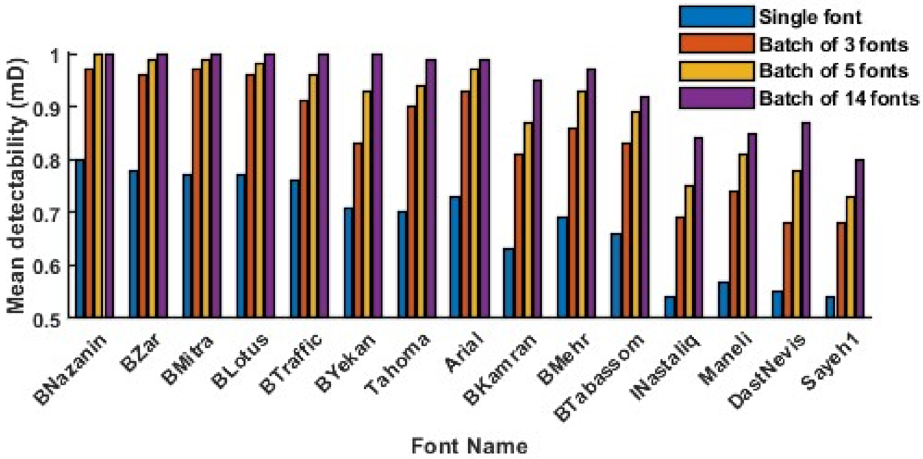

Fig. 12. Mean detectability (mD) of every font using different batches of fonts including batches of single font, three fonts, five fonts, and fourteen other fonts.

The model's performance is further evaluated in Figure 13, which presents the confusion matrix for character recognition of the Sayeh1 font using a model trained on fourteen other fonts. The results indicate that a substantial portion of characters, particularly numbers, mathematical symbols, punctuation marks, and both regular and non-breaking spaces, are accurately recognized. However, challenges remain in distinguishing certain letters. In these instances, the model successfully detects the presence of a character but struggles to differentiate it from visually similar ones. This suggests that further refinements, such as incorporating an OCR post-processing system equipped with a context-aware vocabulary and word-guessing algorithms, could significantly improve character-level accuracy and overall text recognition performance. Notably, this is not exclusive to the Sayeh1 font and is observed across other fonts as well.

To offer a preliminary insight into possible future enhancements, we applied a lightweight post-processing algorithm. While not the central focus of our study, this step demonstrates the potential of such techniques in improving overall recognition performance. For this purpose, a simple spell correction algorithm was employed to refine the detected words and assess the impact of post-processing. 120 Persian words are selected to generate image samples in various fonts. For each font, the model trained on fourteen other fonts is used to detect the words from the image samples. The raw word output produced by the model is then compared against a vocabulary list containing 1,355 Persian words, and replaced with the closest matching word based on similarity. The Character Recognition Rate (CRR) and Word Recognition Rate (WRR) are calculated by comparing the guessed words to the ground truth. CRR represents the proportion of correctly detected characters and is computed by dividing the number of correctly detected characters by the total number of characters in the reference text. WRR, on the other hand, reflects the proportion of words that are correctly identified.

As shown in Table 2, application of spell correction post-processing step significantly improves both CRR and WRR. In comparison to Figure 11, the character-level recognition accuracy is slightly lower than the accuracy expected from the model at first glance. Part of this issue is that the accuracy of character recognition reported in Figure 11 pertains to all 70 characters. However, in fact most errors are related to alphabetic characters, which leads to slightly more mistakes in recognizing letters and words in Table 2. On the other hand, sometimes the model struggles to recognize a frequently occurring letter in Persian words. In this case, although the overall accuracy of character recognition is high, frequent errors in recognizing a single letter can result in many words being misrecognized. Another part of this issue is due to some errors arise during the sequence ordering of recognized letters to guess the word.

Word detection performance is satisfactory for formal and informal fonts. It means that the proposed method can be used for OCR of many other fonts. Interestingly, the detection of words in handwritten-style fonts such as Sayeh1 and DastNevis is also promising, considering the inherent difficulty in recognizing characters from unseen fonts. However, for Maneli and INastaliq fonts, which are characterized by a high degree of character shape overlap, more similar fonts for training and more advanced post-processing algorithms to find the true sequence of letters in the word are required.

Accepted Manuscript: Author Version

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

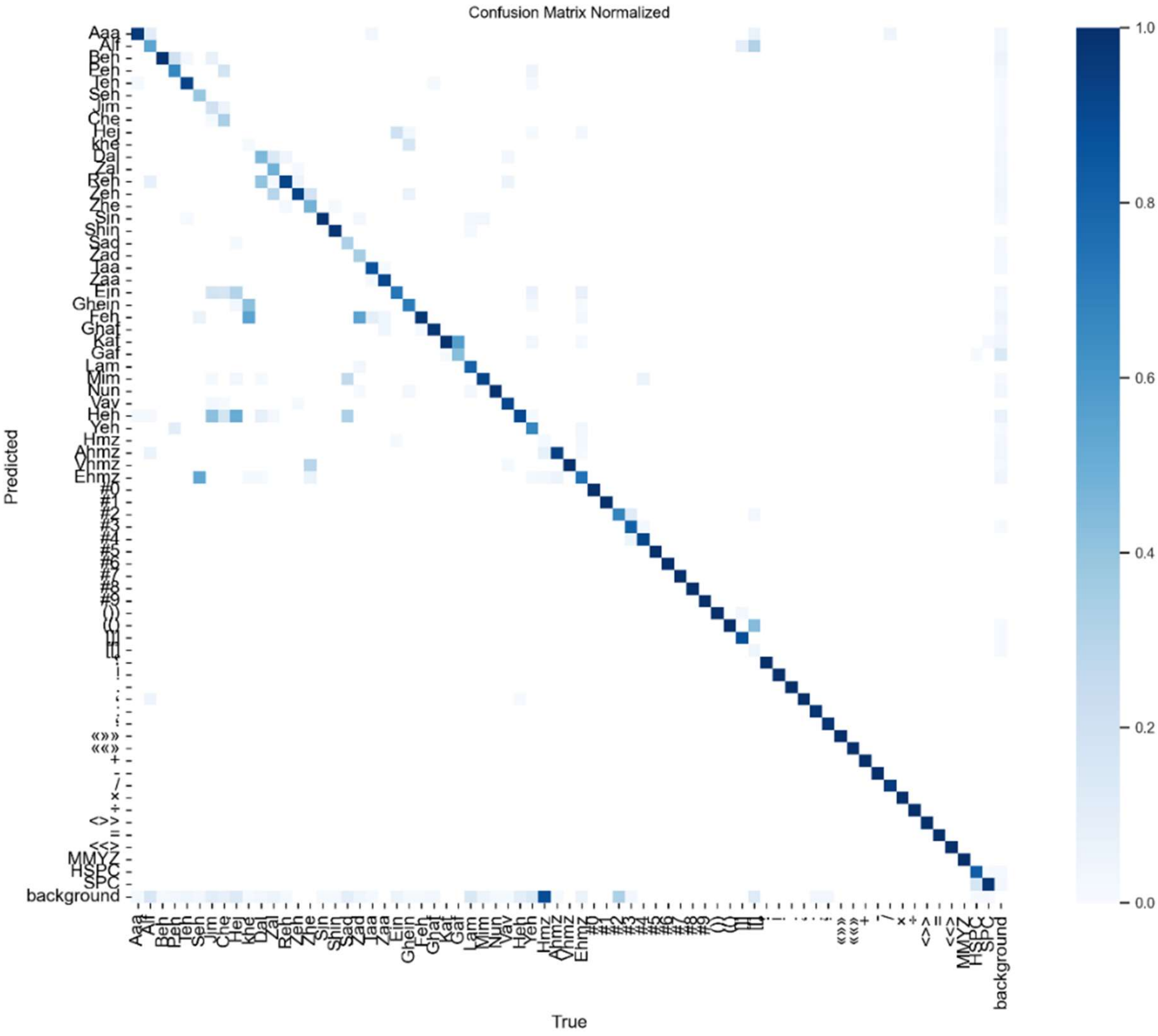

Fig. 13. The confusion matrix for recognizing Sayeh1 characters using a model trained with fourteen other fonts.

Accepted Manuscript

Table 2. Character Recognition Rate (CRR) and Word Recognition Rate (WRR) of the model when trained on fourteen other fonts, before and after spell correction using a vocabulary list.

Font	CRR (%)		WRR (%)	
	Before	After	Before	After
BNazanin	98	99	97	99
BZar	98	99	96	99
BMitra	97	98	97	99
BLotus	96	98	96	99
BTraffic	96	99	92	97
BYekan	88	96	54	92
Tahoma	94	99	75	97
Arial	96	98	86	96
BKamran	92	99	67	97
BMehr	88	95	55	87
BTabassom	85	92	55	84
INastaliq	54	60	05	39
Maneli	66	77	15	61
DastNevis	77	89	29	77
Sayeh1	75	83	27	68

4. CONCLUSION

This study proposes the use of the YOLO object detection model for omni-font Optical Character Recognition

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

(OCR) of Persian scripts, covering formal, informal, and handwritten-style fonts. Synthetic datasets were generated, incorporating words, numbers, mathematical symbols, and punctuation marks. Additionally, a novel approach for detection whitespace characters (regular and non-breaking spaces) was introduced and analyzed, yielding promising results that could potentially eliminate the need for an explicit word segmentation step in OCR systems. In this study, 70 characters of Persian script in 15 different fonts were examined.

Training and validating the YOLO model on individual fonts show that key evaluation metrics approach their final values, nearing 100%, with approximately 700 samples. The confusion matrices confirm that the model accurately detects all characters with minimal misclassification, including small characters, visually similar characters, symmetrical characters, and whitespace characters.

In the following analysis, we explored the possibility of recognizing a new font by training the model on a batch of other fonts. The results indicate that training with multiple fonts enables the recognition of characters from a font that was not included in the training process. Specifically, using batches of three fonts for training improves the mean detectability (mD), surpassing 95% for the most distinguishable formal fonts. Extending the training set to batches of five fonts, further increases mD, exceeding 99% for top formal fonts. Notably, training with fourteen other fonts allows formal fonts to be recognized almost entirely.

For handwritten-style fonts, the results indicate significant improvements in recognition accuracy with exposure to diverse training data. This demonstrates the model's ability to generalize to unseen and visually diverse handwriting-like fonts. For instance, the mD of the Sayeh1 font increased to 80% when the model was trained with fourteen other fonts, while other handwritten-style fonts, such as INastaliq, Maneli, and DastNevis, exhibited even higher recognition performances. The confusion matrix for Sayeh1 shows that while simple characters, such as numbers, mathematical operators, and punctuation marks, are detected with high accuracy, some letters remain challenging to distinguish. However, recognition accuracy could potentially be further enhanced by incorporating vocabulary-based post-processing techniques and word-guessing algorithms.

Overall, this study demonstrates that increasing both the diversity and quantity of training samples significantly enhances OCR performance. While formal fonts exhibit high detectability, further research is required to improve the recognition of handwritten-style fonts and achieve omni-font OCR using object detection models such as YOLO. The proposed method, due to its low resource requirements and flexible structure, is suitable for scaling to large datasets and diverse OCR tasks. The findings suggest that a simple yet highly effective method for handwriting recognition can be developed by constructing a large-scale dataset of natural handwriting samples optimized for object detection frameworks, leveraging ongoing advancements in deep learning techniques and OCR systems.

Data Availability Statement:

The datasets related to this study will be made available upon request from the corresponding author.

Funding Information:

This research was conducted without any external funding.

Accepted Manuscript- Author Version

REFERENCES

[1] H. Noori, **"Persian Car License Plate Recognition using Deep Convolutional Neural Networks,"** *AUT Journal of Electrical Engineering*, vol. 57(2), pp. 295-316, 2025. Available: https://doi.org/10.22060/eej.2024.23296.5601

[2] M. M. Zeinali; S. Ghofrani. **"Application-oriented Farsi license plate recognition using deterministic clustering algorithm and MSER detector,"** *Majlesi Journal of Electrical Engineering*, vol. 10(2), pp. 39-52, 2016. Available: https://mjee.isfahan.iau.ir/article_696233.html

[3] Y. K. Singh, R. Jaiswal, P. Choudhary, B. Chugh, **"Verifying bank checks using deep learning and image processing,"** presented at the *IEEE International Conference on Intelligent Systems for Cybersecurity* (ISCS), pp 1-6, 2024. Available: https://doi.org/10.1109/ISCS61804.2024.10581393

[4] S. Prommas, T. Siriborvornratanakul, **"CNN-based Thai handwritten OCR: an application for automated mail sorting,"** *International Journal of Information Technology*, vol. 16(2), pp. 793-798, 2024. Available: https://doi.org/10.1007/s41870-023-01638-4

[5] C. Q. Lin, D. H. Wang, Y. F. Su, D. W. Ge, X. Y. Zhang, **"OCR4HSV: A Multi-task Learning Approach for Handwritten Signature Verification,"** presented at the *International Conference on Pattern Recognition, Springer, Cham,* pp. 287-302, 2025. Available: https://doi.org/10.1007/978-3-031-78119-3_20

[6] A. Fateh, M. Rezvani, A. Tajary, M. Fateh, **"Persian printed text line detection based on font size,"** *Multimedia Tools and Applications*, vol. 82(2), pp. 2393-2418, 2023. Available: https://doi.org/10.1007/s11042-022-13243-x

[7] M. Aliakbarzadeh, F. Razzazi, **"Online Persian/Arabic Writer Identification using Gated Recurrent Unit Neural Networks,"** *Majlesi Journal of Electrical Engineering*, vol. 14(3), pp. 73-79, 2020. Available: https://doi.org/10.29252/mjee.14.3.9

This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

[8] Z. Khosrobeigi, H. Veisi, E. Hoseinzade, H. Shabanian, **"Persian optical character recognition using deep bidirectional long short-term memory,"** *Applied Sciences*, vol. 12(22), p. 11760, 2022. Available: https://doi.org/10.3390/app122211760

[9] N. Salehian, M. Yazdchi, A. Karimian, **"Farsi Nastaligh Word Recognition by Using Artificial Neural Networks,"** *Majlesi Journal of Electrical Engineering*, vol. 2(4), pp. 1-10, 2008. Available: https://doi.org/10.1234/mjee.v2i4.132

[10] A. K. Bhunia, A. Konwer, A. K. Bhunia, A. Bhowmick, P. P. Roy, U. Pal, **"Script identification in natural scene image and video frames using an attention based convolutional-LSTM network"** *Pattern Recognition*, vol. 85, pp. 172-184, 2019. Available: https://doi.org/10.1016/j.patcog.2018.07.034

[11] H. Hassanpour, N. Samadiani, F. Akbarzadeh, **"A modified self-organizing map neural network to recognize multi-font printed Persian numerals,"** *International Journal of Engineering, Transactions B: Applications*, vol. 30(11), pp. 1700-1706, 2017. Available: https://www.ije.ir/article_73055.html

[12] A. S. Shaker, M. F. Khaleel, O. A. Ismael, R. S. Majeed, M. R. Ahmed, **"Information retrieval system of Arabic alphabetic characters by using hidden Markov Model,"** presented at the *IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-6, 2022. Available: https://doi.org/10.1109/HORA55278.2022.9799843

[13] 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(1), pp. 133-143, 2021. Available: https://doi.org/10.1007/s10032-021-00368-2

[14] B. M. Hasan, Z. J. Jaber, A. A. Habeeb, **"Digits Recognition for Arabic Handwritten through Convolutional Neural Networks, Local Binary Patterns, and Histogram of Oriented Gradients,"** *Baghdad Science Journal*, vol. 21(10), pp. 3322-3322, 2024.

[15] A. A. Sanjrani, J. Baber, M. Bakhtyar, I. Ullah, M. S. Naveed, W. Noor, A. Basit, A. Khan, N. Sheikh, **"Extended framework for Sindhi numerals OCR using gradient orientation histograms,"** *Journal of Intelligent & Fuzzy Systems*, vol. 43(2), pp. 2045-2056, 2022. Available: https://doi.org/10.3233/JIFS-219304

[16] J. Wang, **"A study of the ocr development history and directions of development,"** *Highlights in Science, Engineering and Technology*, vol. 72, pp. 409-415, 2023. Available: https://doi.org/10.54097/bm665j77

[17] A. M. Al-Shatnawi, F. Al-Saqqar, **"A. Souri, Arabic handwritten word recognition based on stationary wavelet transform technique using machine learning,"** *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21(3), pp. 1-21, 2021. Available: https://doi.org/10.1145/3474391

[18] A. Kumar, B. Murugan, **"Handwritten Digit Recognition Using Neural Network with Gabor Filter for Information Fusion,"** presented at the *International Conference on Machine Learning and Big Data Analytics, Cham: Springer International Publishing*, pp. 411-421, 2022. Available: https://doi.org/10.1007/978-3-031-15175-0_34

[19] V. M. Ashiq, E. J. T. Fredrik, **"An OCR for Arabic character recognition with advanced principal component analysis based on feature extraction and fuzzy-KNN based classification,"** *International Journal of Health Sciences*, vol. 6(S1), pp. 12205–12224, 2022. Available: https://doi.org/10.53730/ijhs.v6nS1.7918.

[20] M. Mohammadpoor; A. Mehdizadeh; H. A. Noghabi. **"A Novel Method for Persian Handwritten Digit Recognition Using Support Vector Machine,"** *Majlesi Journal of Electrical Engineering*, vol. 12(3), pp. 63-67, 2018. Available: https://mjee.isfahan.iau.ir/article_696315.html

[21] V. Moradi, F. Razzazi, A. Behrad. **"Recognition of Handwritten Persian Two-digit Numerals Using a Novel Hybrid SVM/HMM algorithm,"** *Majlesi Journal of Electrical Engineering*, vol. 10(3), pp. 19-25, 2016. Available: https://mjee.isfahan.iau.ir/article_696236.html

[22] N. M. Alharbi, A. H. Osman, A. A. Mashat, H. J. Alyamani, **"Letter Recognition Reinvented: A Dual Approach with MLP Neural Network and Anomaly Detectionm,"** *Computer Systems Science & Engineering*, vol. 48(1), pp. 175-198, 2024. Available: https://doi.org/10.32604/csse.2023.041044

[23] S. Alghyaline, **"Arabic Optical Character Recognition: A Review,"** *CMES-Computer Modeling in Engineering & Sciences*, vol. 135(3), pp. 1825-1861, 2023. Available: https://doi.org/10.32604/cmes.2022.024555

[24] M. Elleuch, R. Maalej, M. Kherallah, **"A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition,"** *Procedia Computer Science*, vol. 80, pp. 1712-1723, 2016. Available: https://doi.org/10.1016/j.procs.2016.05.512

[25] B. Alizadehashraf, S. Roohi, **"Persian handwritten character recognition using convolutional neural network,"** presented at 10th *Iranian Conference on Machine Vision and Image Processing (MVIP)*, *IEEE*, pp. 247-251, 2017. Available: https://doi.org/10.1109/IranianMVIP.2017.8342359

[26] M. Aarif K.O, S. Poruran, **"OCR-nets: variants of pre-trained CNN for Urdu handwritten character recognition via transfer learning,"** *Procedia computer science*, vol. 171, pp. 2294-2301, 2020. Available: https://doi.org/10.1016/j.procs.2020.04.248

[27] S. Khosravi, A. Chalechale, **"Chimp optimization algorithm to optimize a convolutional neural network for recognizing Persian/Arabic handwritten words,"** *Mathematical Problems in Engineering*, vol. 2022(1), p. 4894922, 2022. Available: https://doi.org/10.1155/2022/4894922

[28] A. Naseer, K. Zafar, **"Meta features-based scale invariant OCR decision making using LSTM-RNN,"** *Computational and Mathematical Organization Theory*, vol. 25, pp. 165-183, 2019. . Available: https://doi.org/10.1007/s10588-018-9265-9

[29] M. R. Soheili, M. R. Yousefi, E. Kabir, D. Stricker, **"Merging clustering and classification results for whole book recognition,"** presented at 10th *Iranian Conference on Machine Vision and Image Processing (MVIP)*, *IEEE*, pp. 134-138, 2017. Available: https://doi.org/10.1109/IranianMVIP.2017.8342338

[30] S. M. Mousavi, V. Lyashenko, **"Extracting old persian cuneiform font out of noisy images (handwritten or inscription),"** presented at 10th *Iranian Conference on Machine Vision and Image Processing (MVIP)*, *IEEE*, pp. 241-

Accepted Manuscript - Author Version

 This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

246, 2017. Available: https://doi.org/ 10.1109/IranianMVIP.2017.8342358

[31] R. Mondal, S. Malakar, E. H. Barney Smith, R. Sarkar, **"Handwritten English word recognition using a deep learning based object detection architecture,"** *Multimedia Tools and Applications*, vol. 81(1), pp. 975-1000, 2022. Available: https://doi.org/10.1007/s11042-021-11425-7

[32] A. A. Demir, U. Ozkaya, **"Ottoman character recognition on printed documents using deep learning,"** *Mühendislik Bilimleri ve Tasarım Dergisi*, vol. 12(2), pp. 392-402, 2024. Available: https://doi.org/10.21923/jesd.1383926

[33] A. Tourani, S. Soroori, A. Shahbahrami, A. Akoushideh, **"Iranis: A large-scale dataset of Iranian vehicles license plate characters,"** presented at 5th *International Conference on Pattern Recognition and Image Analysis (IPRIA)*, *IEEE*, pp. 1-5, 2021. Available: https://doi.org/10.1109/IPRIA53572.2021.9483461

[34] K. Hemanth, T. J. Nagalakshmi, **"Improving Accuracy of Face Detection in ID Proofs using CNN and Comparing with DLNN,"** presented at 8th *International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, *IEEE*, pp. 1-6, 2023. Available: https://doi.org/10.1109/ICONSTEM56934.2023.10142926

[35] M. Mosannafat, F. Taherinezhad, H. Khotanlou, E. Alighardash, **"Farsi text detection and localization in videos and images,"** presented at 9th *Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), IEEE*, pp. 1-6, 2022. Available: https://doi.org/10.1109/CFIS54774.2022.9756472

[36] M. Gandomkar, S. Khoramipour, **"Optical Character Recognition (OCR) in Cursive Scripts Using Object Detection Networks,"** *Tabriz Journal of Electrical Engineering*, (Accepted 2024). Available: https://doi.org/10.22034/tjee.2024.62945.4877

[37] S. Hatami, S. Behnam, R. Shamsaee, **"Improving Detection of Capsule Endoscopy Using YOLO,"** *Tabriz journal of electrical engineering,* vol. 57(2), pp. 295-316, 2024. Available: https://doi.org/10.22034/tjee.2024.58239.4711

[38] R. Mondal, S. Malakar, E. H. Barney Smith, R. Sarkar, **"Handwritten English word recognition using a deep learning based object detection architecture,"** *Multimedia Tools and Applications*, vol. 81(1), pp. 975-1000, 2022. Available: https://doi.org/10.1007/s11042-021-11425-7

[39] S. Alghyaline, **"A printed arabic optical character recognition system using deep learning,"** *Journal of Computer Science*, vol. 18(11), pp. 1038-1050, 2022. Available: https://doi.org/10.3844/jcssp.2022.1038.1050

Accepted Manuscript: Author version

 This article has license CC BY 4.0 https://creativecommons.org/licenses/by/4.0/