



# Improving Persian Handwritten Digit Recognition using Convolutional Neural Network

H. Noori \*<sup>1</sup>

<sup>1</sup>Department of Electrical of Engineering, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.

---

## ABSTRACT

Persian digit recognition plays a crucial role in computer vision and pattern recognition. Existing algorithms fall into two categories: traditional methods and deep learning approaches. While many deep learning techniques are documented, they often depend on pre-trained networks with numerous parameters, requiring substantial resources and time for training and prediction. This paper presents a novel convolutional neural network (CNN) architecture for Persian digit recognition that is shallower than current models, thereby reducing the number of trainable parameters. We introduce dilated convolution layers to capture larger features without increasing parameters and propose a combined loss function to improve accuracy. Trained on the HODA dataset, our method achieves a validation accuracy of 99.82%, test accuracy of 99.79%, and training accuracy of 100%. The proposed network demonstrates enhanced accuracy, faster performance, and significantly reduced implementation time due to its streamlined architecture.

---

## ARTICLE INFO

### *Article history:*

Research paper

Received 02, April 2024

Accepted 03, June 2024

Available online 03, August 2024

---

*Keyword:* Digit recognition, deep learning, handwritten recognition.

---

AMS subject Classification: 62P30.

---

\*Corresponding author: H. Noori. Email: [h.noori@vru.ac.ir](mailto:h.noori@vru.ac.ir)

## 1 Introduction

Optical Character Recognition (OCR) is a highly engaging research topic in computer vision and artificial intelligence. It involves converting text written by humans or machines into a sequence of characters that can be read by a computer or processor [1]. OCR has numerous applications, including car license plate recognition, reading cheque amounts in banking systems, processing postal codes, and digitizing old documents. It can be divided into two categories: digit recognition and letter recognition. This paper focuses specifically on handwritten digit recognition.

While there is an extensive body of literature on English character recognition, Persian and Arabic handwritten recognition remains an open challenge. Numerous countries, including Iran, Yemen, Oman, and Saudi Arabia, use Persian or Arabic characters, collectively comprising a population of over 700 million people worldwide. Therefore, it is essential to develop a fast and reliable method for recognizing Persian and Arabic handwritten characters.

Several factors contribute to the challenges of handwritten recognition in Persian and Arabic texts. One significant factor is that Persian digits can be written in various shapes. Fig. 1 illustrates various samples of Persian digits. As shown in the figure, the digit '0' can appear in two different forms: with a hole and without a hole. Additionally, the digits '2', '3', '4', '5', and '6' also have two distinct forms. Furthermore, the variability in handwriting styles among different individuals presents another challenge for recognition. As shown in Figure 2, the left column depicts a '0' and a '5', which can easily be confused due to their similar appearance. Similarly, the right column displays a '6' and an '8', which are also quite difficult to distinguish from one another.



Figure 1: Different shape of Persian digits.

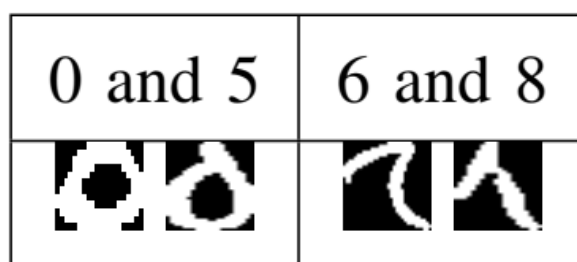


Figure 2: Different handwritings by different persons.

OCR can be performed online and offline. Online OCR is suitable for applications such as car entry gates, while offline OCR is ideal for processing old handwritten documents. Consequently, developing a method that offers high accuracy and speed is essential. In this paper, we propose a new method that achieves both fast performance and high accuracy.

The remainder of this paper is organized as follows: Section 2 presents a literature review. Next, Section 3 explains the dataset and proposes a new architecture. Simulation results and implementation details are discussed in Section 4, and finally, Section 5 concludes the paper.

## 2 Related Works

In general, OCR algorithms can be categorized into two main types: traditional approaches [1]- [12] and deep learning-based methods [13]- [27]. Traditional algorithms involve extracting features defined by the user, which are then fed into a classifier, such as a support vector machine (SVM), to recognize digits. In contrast, deep learning-based methods typically utilize artificial neural networks—especially convolutional neural networks—to automatically extract features and classify the input digits based on those features. In this paper, we focus on deep learning-based approaches.

In [7], the authors aimed to reduce the size of the feature vector by employing principal component analysis (PCA). The obtained feature vector was then classified using a SVM, resulting in an accuracy of 99.07%. Sadri et al. [8] utilized a gradient histogram method in combination with a multilayer perceptron neural network for digit recognition, achieving an accuracy of 98.57%. In [14], the authors implemented a vector symbolic representation as a layer within a neural network. Although the designed network proved to be robust to noise and fast, it reached a maximum accuracy of 88.9% for character recognition, which does not extend to digit recognition.

In [9], three approaches are proposed for Persian digit recognition. The first approach involves utilizing geometric and correlation-based features, achieving an accuracy of 99.3% on the HODA dataset. Additionally, the authors employed two deep learning-based methods for recognizing Persian handwritten digits. The first method used an auto-encoder with three convolutional layers, while the second employed a convolutional neural network (CNN) comprising 15 layers, resulting in a maximum accuracy of 99.54%. However, it is important to note that this CNN has a large number of parameters and demonstrates lower accuracy compared to other deep learning-based methods.

In [10], the author initially aimed to collect a real dataset for Persian words, named PHOND, but employed the HODA dataset for digit recognition. Two pre-processing phases were implemented on the collected dataset. In the first phase, the images were converted to binary format, followed by segmenting a string of digits (i.e., a number) into individual digits. In the second preprocessing phase, the images were resized to  $80 \times 80$  pixels. The author used a modified framing feature as the feature selection step in her algorithm. Ultimately, the features were input into a SVM and a k-nearest neighbor (KNN) classifier, achieving an accuracy of 99.07%. In [11], the authors employed a Bag of Visual Words (BoVW) technique for feature selection. They then used a scale-invariant feature transform (SIFT) to describe the features. The resulting feature vectors were classified using three different classifiers: a multilayer perceptron (MLP), SVM, and a quantum neural network (QNN). The research demonstrated that using SIFT followed by

the QNN achieved the highest accuracy of 99.30%.

Khorashadizadeh et al. [12] proposed a method that combines four directional chain code histograms (CCH) and histograms of oriented gradients (HOG) to create a feature set. A SVM with a radial basis function kernel was used to classify these features. The proposed method achieved an accuracy of 99.58% through 5-fold cross-validation on the HODA dataset. However, the authors employed two feature selection methods that have high computational complexity. Additionally, it is important to note that using K-fold validation can lead to information leakage from the validation data, which may result in the classifier learning from the validation data in a manner similar to the training data.

In [15], a phone number recognition system is proposed in which, after segmentation, the numbers are processed by a six-layer CNN, achieving an accuracy of 99.34%. In [16], a nine-layer CNN is introduced for detecting Persian handwritten characters, which includes four fully connected layers and attained an accuracy of 99.7%. Although this paper focuses on character recognition rather than digit recognition, it is important to note that fully connected layers contain many parameters that need to be trained. As a result, while the proposed method achieves good accuracy, it requires a significant number of parameters, leading to high computational overhead for the processor using this network.

In [17], the authors employed a CNN for automatic feature selection and, ultimately, utilized a SVM as the classifier instead of a fully connected layer, thereby reducing computational complexity. The algorithm reported a maximum accuracy of 99.56%. In [18], the authors utilized a pre-trained network, AlexNet [19], and subsequently retrained it with data augmentation on the HODA dataset. The network described in [18] achieved an accuracy of 99.67%.

In [20], Bossaghzadeh proposed two deep learning-based approaches for Persian digit recognition. In the first method, he employed VGGNet to extract features, which were then input to a SVM. In the second method, to reduce computational overhead, he used PCA to decrease the feature dimensionality before feeding the resulting features into a linear SVM. This paper achieved a maximum accuracy of 99.69%. The author in [20] utilized VGGNet, which requires an input size of  $224 \times 224$ , but he used the HODA dataset, where the image size is at most  $32 \times 32$ . Consequently, he needed to enlarge the dataset samples, which would involve generating information that is not present in the original data. As a result, their findings are based on artificially augmented data rather than authentic data.

In [21], the authors utilized CNN and recursive neural networks, specifically bidirectional long short-term memory (BLSTM), for feature extraction in Persian digit recognition. They ultimately employed a stacking ensemble classifier to enhance detection accuracy. The proposed method achieved an accuracy of 99.39% on the HODA dataset.

In [22], a new network architecture was proposed to detect handwritten digits from digit strings in historical texts. The authors first introduced a novel dataset based on historical documents. They designed a two-stage network for digit detection and recognition: the first stage selects a digit from the string, while the second stage recognizes the selected digit. The proposed network, named DIGINET, consists of three residual units, each containing three CNNs followed by an addition and a Batch-normalization layer. Addi-

Table 1: Number of samples in each category in HODA dataset

Category	0	1	2	3	4	5	6	7	8	9
Number of samples	10070	10330	9923	10334	10333	10110	10254	10363	10264	10371

tionally, six convolutional layers are employed to extract features, along with eight more CNN layers dedicated to digit recognition. DIGINET achieved a maximum accuracy of 97.12%.

In [23], the authors employed two distinct architectures, DenseNet and Xception, to detect digits, characters, and words. They utilized pre-trained models of DenseNet121, DenseNet161, and DenseNet169, achieving a recognition rate of 99.72% for digit detection. However, these networks, with their numerous layers and significant number of parameters, require substantial computational resources for training, such as extensive convolution operations, which can result in lower processing speeds. In [28], a novel algorithm based on probabilistic neural networks (PNN) and MLP neural networks is proposed for classifying digits in the HODA dataset.

Deep convolutional neural networks (CNNs) have achieved remarkable success in various computer vision tasks, with network architecture being a pivotal factor in performance. This paper introduces a novel CNN architecture that prioritizes both accuracy and computational efficiency. By incorporating dilated convolutions, the proposed model effectively expands receptive fields without increasing model depth, leading to improved accuracy and reduced computational overhead. Furthermore, a combined loss function is employed to enhance overall performance. Experimental results on the HODA dataset [29] demonstrate the efficacy of our approach, achieving a validation accuracy of 99.82% and a test accuracy of 99.79%.

## 3 Method

### 3.1 Dataset Explanation

In this paper, we leverage the HODA dataset for training and testing our model. The HODA dataset consists of handwritten Farsi digits collected from B.Sc. student registration forms in Iran. The images are scanned at a resolution of 200 dots per inch (dpi). Fig. 3 illustrates examples of different writing styles for digits 0 to 9, from left to right. Fig. 4 could showcase the variation in image quality within the dataset. Unlike commonly used datasets with fixed image sizes, the HODA dataset features images of varying dimensions. The dataset comprises a total of 102,352 samples. We allocate 60,000 and 20,000 samples for training and testing purposes, respectively. An additional 22,352 samples are available for validation. Table 1 details the number of samples in each digit category. For further details on the HODA dataset, readers can refer to [29].



Figure 3: Samples with different writing styles in the HODA dataset

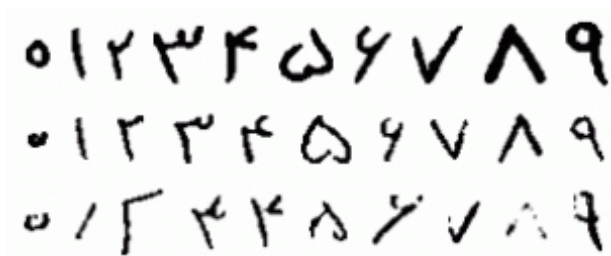


Figure 4: Samples with different qualities in the HODA dataset

### 3.2 Pre-processing

The HODA dataset contains images of varying dimensions. To determine appropriate input dimensions for our network, the average height and width of all images are calculated, resulting in 24 rows and 28 columns. All images were subsequently resized to a uniform size of (24, 28) pixels. For data normalization, pixel values were converted to float32 and scaled by dividing each image by its maximum pixel value (255). The preprocessed images were then organized into arrays with dimensions (60000, 24, 28, 1) for the training set and (20000, 24, 28, 1) for the testing set.

### 3.3 Network Architecture

As discussed in the previous section, existing Persian digit recognition algorithms exhibit two primary shortcomings. Firstly, state-of-the-art models often demand substantial computational resources due to their complex and very deep architectures. For instance, DenseNet121, DenseNet161, and DenseNet169, employed in [23], incorporate at least 121 layers, contrasting with the shallower architectures commonly used for English digit recognition. Secondly, these models frequently utilize input dimensions considerably larger than the dataset's image size, prompting the network to learn from interpolated data rather than genuine image features. To illustrate, DenseNet121 [23] expects inputs of size (224, 224, 3), whereas HODA dataset images measure (32, 32, 3), implying that DenseNet121 is trained on interpolated data while there are several English digit datasets with larger

Table 2: Hyper parameters of green section

Layer	kernel size	# of kernel	stride	dilation	input size	output size
Layer 1	3	16	1	1	$24 \times 28 \times 1$	$24 \times 28 \times 16$
layer 2	3	32	1	1	$24 \times 28 \times 16$	$24 \times 28 \times 32$
layer 3	3	64	1	1	$24 \times 28 \times 32$	$24 \times 28 \times 64$
layer 4	3	128	1	1	$24 \times 28 \times 64$	$24 \times 28 \times 128$

Table 3: Hyper parameters of red section

Layer	kernel size	# of kernel	stride	dilation	input size	output size
Layer 1	3	16	1	2	$24 \times 28 \times 1$	$24 \times 28 \times 16$
layer 2	3	32	1	2	$24 \times 28 \times 16$	$24 \times 28 \times 32$
layer 3	3	64	1	2	$24 \times 28 \times 32$	$24 \times 28 \times 64$

size samples.

To address the aforementioned shortcomings, we propose a novel convolutional neural network architecture for Persian handwritten digit recognition. This architecture incorporates larger kernel sizes to enhance feature extraction while employing dilated convolutions to mitigate computational complexity. As illustrated in Fig. 5, the proposed model utilizes a multi-branch (multi-scale: parallel green, red and yellow layers in fig. 5) structure, wherein the input image is processed by three parallel convolutional paths. The resulting multi-scale feature maps are subsequently concatenated with the original image to enrich the feature representation (brown layer in fig. 5). This design enables the network to effectively capture both low-level and high-level image features. Subsequent layers extract higher-level features.

To enhance digit recognition accuracy, larger features are essential. While larger kernels can capture these features, they introduce excessive parameters. To address this, we propose replacing traditional convolution layers with dilated convolution layers [33]. This approach enables the extraction of larger features without increasing parameter count. Dilated convolutions are integrated into the parallel sections of our network (green, red, and yellow layers in fig. 5). As illustrated in Fig. 6, dilated convolutions with dilations of 2 and 3 (fig. 6-(b) and 6-(c)) capture  $5 \times 5$  and  $7 \times 7$  features, respectively, using the same number of parameters as a standard  $3 \times 3$  convolution 6-(a).

The top branch of the network’s parallel section (green layers in fig. 5) consists of four two-dimensional  $3 \times 3$  standard convolution layers that preserve input dimensions. Table 2 summarizes the hyperparameters for this branch.

The middle branch (red layers in fig. 5) employs three two-dimensional  $3 \times 3$  dilated convolution layers with a dilation rate of 2, maintaining the input dimensions. Table 3 details the hyperparameters for this branch.

Similarly, the bottom branch (yellow layers in fig. 5) consists of three two-dimensional  $3 \times 3$  dilated convolution layers with a dilation rate of 3, preserving input size. Table 4

Table 4: Hyper parameters of yellow section

Layer	kernel size	# of kernel	stride	dilation	input size	output size
Layer 1	3	16	1	3	$24 \times 28 \times 1$	$24 \times 28 \times 16$
layer 2	3	32	1	3	$24 \times 28 \times 16$	$24 \times 28 \times 32$
layer 3	3	64	1	3	$24 \times 28 \times 32$	$24 \times 28 \times 64$

Table 5: Hyper parameters of concatenation layer

Layer in/out	size
Input Size	$(24 \times 28 \times 1), (24 \times 28 \times 128), (24 \times 28 \times 64), (24 \times 28 \times 64)$
Output size	$(24 \times 28 \times 257)$

outlines the hyperparameters for this branch.

Following the parallel section, a concatenation layer (brown in fig. 5) stacks feature maps from the parallel branches with the original input image. The combined output is then fed to subsequent layers for higher-level feature extraction. Essentially, this module separates extracted features from the raw input before passing the remaining information onward. Table 5 outlines the layer’s properties.

The concatenated features are fed into three two-dimensional  $3 \times 3$  convolutional layers (orange in fig. 5). Each orange layer comprises a  $3 \times 3$  convolution followed by a  $2 \times 2$  max-pooling layer to downsample feature maps and reduce computational cost (fig. 7). Table 6 details the hyperparameters for these orange layers.

Extracted features are flattened into a vector using a Flatten layer (gray in fig. 5, parameters in Table 7). This vector is then fed into the classifier section (purple layers in fig. 5). Each purple layer consists of a fully connected layer followed by a Dropout layer to mitigate overfitting (fig. 8). Finally, the output of these layers is passed to a Softmax layer for 10-class classification (parameters in 8). The proposed network comprises 24 layers, including input and output layers, with a total of 848,666 trainable parameters.

Table 6: Hyper parameters of orange section

Layer type	kernel size	# of kernel	stride	dilation	input size	output size
Convolution	3	64	1	1	$24 \times 28 \times 257$	$22 \times 26 \times 64$
Maxpooling	2	-	1	-	$22 \times 26 \times 64$	$11 \times 13 \times 64$
Convolution	3	128	1	1	$11 \times 13 \times 64$	$9 \times 11 \times 128$
Maxpooling	2	-	1	-	$9 \times 11 \times 128$	$4 \times 5 \times 128$
Convolution	3	256	1	1	$4 \times 5 \times 256$	$2 \times 3 \times 256$
Maxpooling	2	-	1	-	$2 \times 3 \times 256$	$1 \times 1 \times 256$



Table 7: Hyper parameters of flatten layer

Layer in/out	size
Input Size	$(1 \times 1 \times 256)$
Output size	$(1 \times 256)$

Table 8: Hyper parameters of dense section

Layer type	# of neurons	input size	output size	Dropout (%)
Dense	500	$1 \times 256$	$1 \times 500$	-
Dropout	-	$1 \times 500$	$1 \times 500$	0.2
Convolution	100	$1 \times 500$	$1 \times 100$	-
Dropout	-	$1 \times 100$	$1 \times 100$	0.3
Convolution	10	$1 \times 500$	$1 \times 10$	-

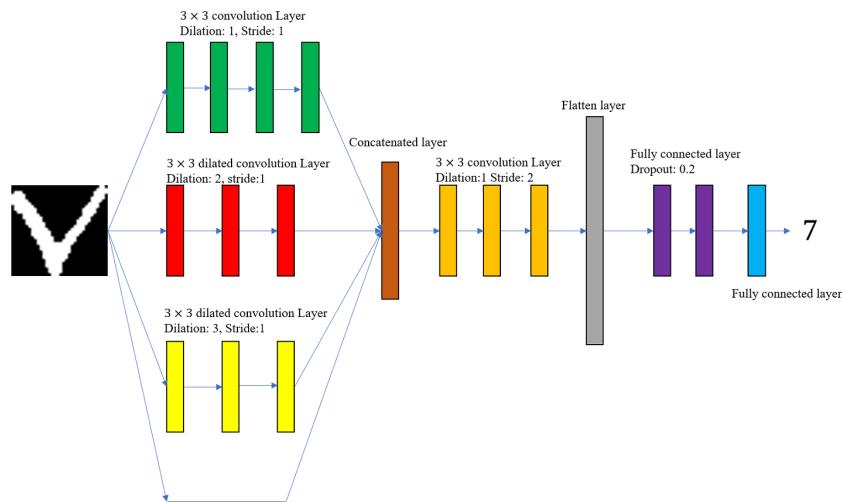


Figure 5: The proposed network architecture

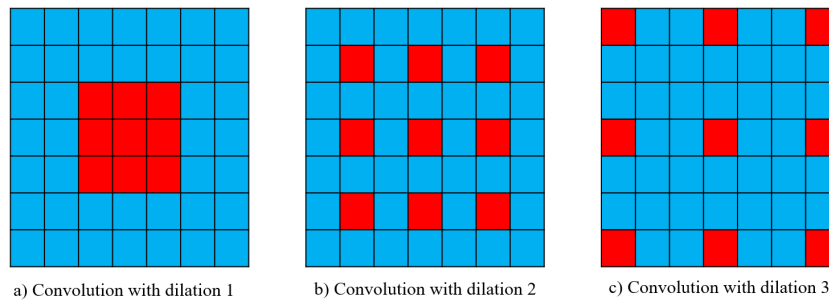


Figure 6: Difference between convolution layers with different dilations

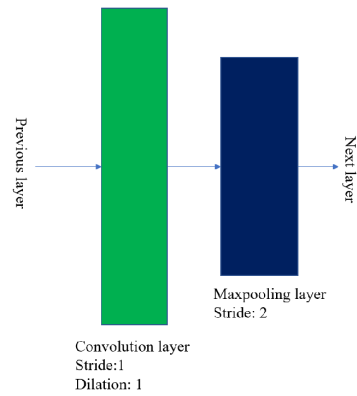


Figure 7: A convolution layer with stride 2 in fig. 5 is equivalent to an ordinary convolution layer followed by a maxpooling layer with stride 2

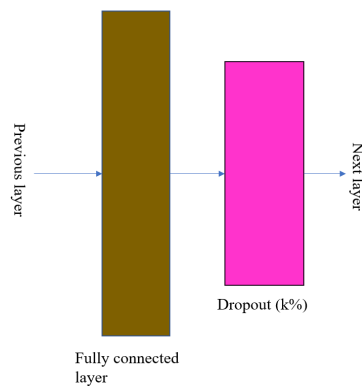


Figure 8: Each purple layer in fig 5 is equivalent to a Fully connected layer followed by a dropout layer

## 4 Training Detail and Experimental Results

This section first outlines the training methodology, including activation functions and learning parameters. Subsequently, the implementation results are presented.

To enhance the proposed network’s accuracy, a hybrid loss function is employed. Early in training, the error between network output and ground truth is typically large and gradually diminishes. To emphasize error correction in later training stages, we combine mean squared error (MSE) for early epochs with absolute error (AE) for later epochs. The proposed loss function is defined as follows:

$$L = \alpha \frac{1}{N} \sum_{i=1}^N (O_n - O_d)^2 + \beta \frac{1}{N} \sum_{i=1}^N |O_n - O_d| \quad (1)$$

Here,  $L$  denotes the loss function,  $N$  is the number of output categories ( $N = 10$  in this case),  $O_n$  represents the network output,  $O_d$  is the desired output, and  $|\cdot|$  signifies absolute value. The hyperparameters  $\alpha$  and  $\beta$  regulate the error contribution. Initially set to  $\alpha = 0.8$  and  $\beta = 0.2$ , these values are updated as follows:

$$\alpha_{i+1} = \alpha_i - (i \times M) \quad (2)$$

$$\beta_{i+1} = \beta_i + (i \times M) \quad (3)$$

Where  $\beta_i$  represents the value of  $\beta$  at epoch  $i$ , and  $M = 0.001$  is the step size for adjusting these parameters. This approach gradually diminishes the influence of mean squared error while amplifying the impact of absolute error as the training progresses.

Rectified Linear Units (ReLU) are employed as the activation function for all layers except the output layer. Softmax is utilized in the output layer to generate a probability distribution over the digits 0 to 9. The network is optimized using the "Adam" optimizer with a learning rate of 0.0001. Training is conducted with a batch size of 256 for 500 epochs, and 10% of the data is reserved for validation. Fig. 9 depicts the training accuracy, which reaches 100%. Fig. 10 illustrates the validation accuracy, peaking at 99.82%. The model achieves a test accuracy of 99.79%, surpassing the 99.76% reported in [23] despite having significantly fewer layers (24 compared to at least 161 in [23]).

To assess the proposed method’s performance, a confusion matrix for handwritten Persian digit recognition is presented in Table 9. The results indicate perfect recognition of digit 1 and exceptionally high accuracy for digit 8.

To benchmark the proposed method, Table 10 presents the accuracy of several existing algorithms. Notably, few studies report network parameter counts or inference time, limiting direct comparison. However, [23] provides such data for various architectures, with DenseNet121 achieving the highest accuracy. This model contains 6.8 million parameters and requires 0.434 seconds per prediction. In contrast, our proposed network with 848,666 parameters attains a prediction time of 17 milliseconds on a Tesla T4 GPU using Google Colab. Despite fewer parameters and significantly lower computational cost, our approach

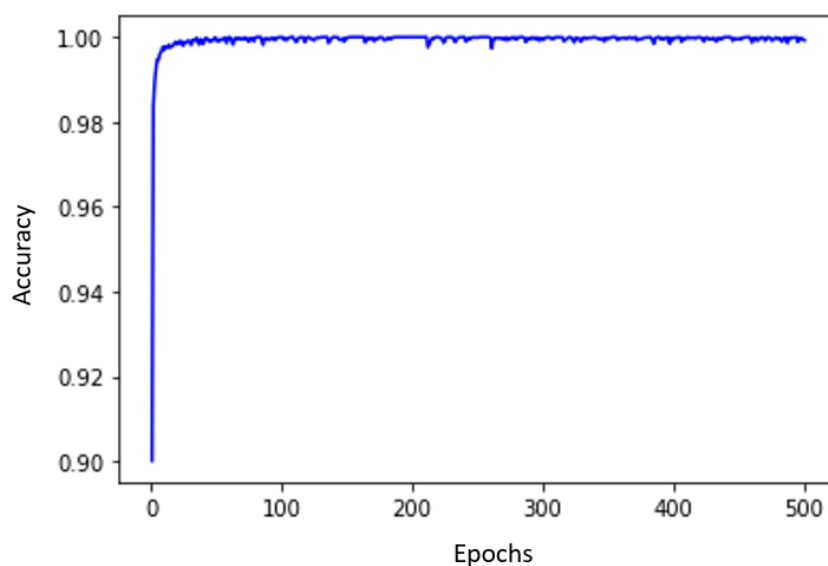


Figure 9: The accuracy of the proposed network in the training step

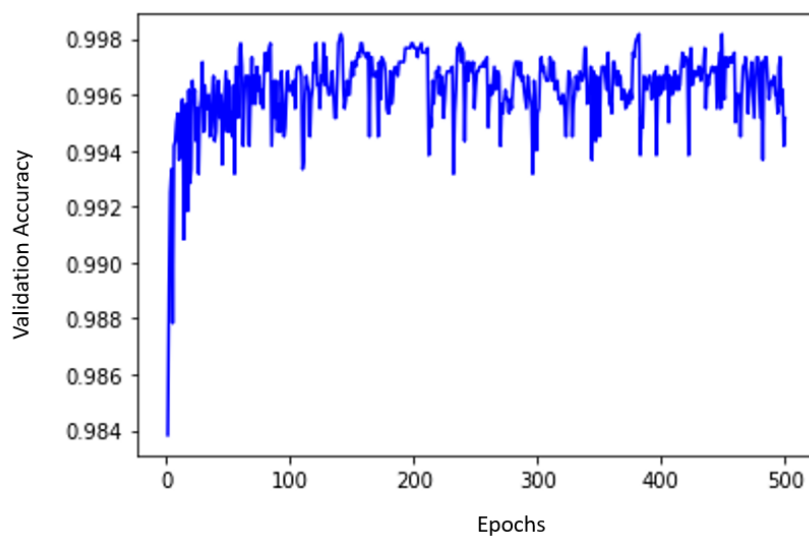


Figure 10: The validation accuracy of the proposed network in the training step

Table 9: Confusion matrix for handwritten digits recognition

True labels ↓/Predicted labels →	0	1	2	3	4	5	6	7	8	9
0	1994	0	0	1	0	2	1	1	1	0
1	0	2000	0	0	0	0	0	0	0	0
2	0	2	1995	1	1	0	0	0	1	0
3	0	0	1	1995	1	0	1	0	1	1
4	0	0	1	3	1996	0	0	0	0	0
5	4	1	1	0	1	1993	0	0	0	0
6	0	1	4	0	0	0	1995	0	0	0
7	0	0	0	0	0	0	1	1996	1	2
8	0	1	0	0	0	1	0	0	1998	0
9	1	0	0	0	1	1	1	0	0	1996

Table 10: Comparison of the proposed network with the state of art alternatives in terms of correctly recognizing handwritten Persian digits on HODA dataset.

Reference number	Accuracy %	Prediction time (ms)
Ref. [2]	99.37	240
Ref. [23]	99.72	434
Ref. [7]	99.07	35000
Ref. [8]	98.57	27000
Ref. [9]	99.54	30000
Ref. [15]	99.34	15
Ref. [16]	99.70	18
Ref. [17]	99.56	12
Ref. [18]	99.67	17
Ref. [20]	99.69	320
Ref. [21]	99.39	240
Ref. [22]	97.12	450
Proposed	<b>99.82</b>	<b>9</b>

outperforms DenseNet121 in terms of accuracy. This improvement can be attributed to the effective use of larger kernels and the carefully designed network architecture.

To ensure a fair comparison, all algorithms listed in Table 10 were implemented in Jupyter Notebook and trained on Google Colab using the HODA dataset. The trained models and weights were then transferred to a laptop equipped with an Intel Core i7-10750H 2.6GHz CPU, 32GB RAM, and a GeForce RTX 2070 Max-Q design GPU for testing and performance evaluation. The resulting implementation times are reported in Table 10. As shown in the table, the proposed method achieves the highest accuracy while being the fastest.

Table 11 presents a comparison of the recall criteria between the proposed method and state-of-the-art techniques. The proposed approach consistently outperforms the others, achieving higher recall rates in five out of the evaluated cases. Similarly, Table 12 shows a precision comparison, where the proposed method again surpasses the competitors in five instances.

For a more comprehensive comparison, Table 13 presents the F-score of the proposed and existing algorithms. The results indicate that our method outperforms the others in 70% of digit recognition cases.

For further analysis, fig. 11 presents examples of images where the proposed algorithm misclassified digits. Notably, even human observers would struggle to accurately label these images, suggesting that the network’s performance is comparable to human capabilities in challenging conditions. To evaluate the impact of optimization methods, fig. 12 compares the maximum accuracy achieved by the proposed algorithm using different optimizers. The results indicate that the ‘Adam’ optimizer yields the best performance.

Table 11: Comparison of recall criteria between the proposed method and the state-of-the-art algorithms.

Label/Method	Ref. [2]	Ref. [4]	Ref. [10]	Ref. [7]	Ref. [11]	Ref. [1]	Ref. [24]	Ref. [21]	Proposed
0	97.8	98.6	99.22	99.05	99.5	98.2	98.5	99.7	<b>99.7</b>
1	99.3	97.8	99.89	99.55	99.9	99.4	99.85	99.8	<b>100</b>
2	99.05	99.4	98.78	98.10	98.2	98.8	98.85	99.45	<b>99.75</b>
3	95.7	91.8	97.16	97.70	98.5	98.8	98.8	98.5	<b>99.75</b>
4	98.95	95.8	98.54	<b>100</b>	99.4	99.6	99.3	99.35	99.8
5	99.35	95.8	99.32	99.45	99.5	99.0	<b>99.8</b>	99.55	99.65
6	98.7	95.0	99.10	98.95	99.2	99.3	99.65	99.3	<b>99.75</b>
7	99.6	<b>100</b>	99.70	98.90	99.6	99.6	99.85	99.4	99.8
8	99.85	<b>100</b>	99.89	<b>100</b>	<b>100</b>	99.4	99.95	99.75	99.9
9	98.8	99.2	99.16	<b>100</b>	99.5	99.2	99.2	99.05	99.8

Table 12: Comparison of precision criteria between the proposed method and the state-of-the-art algorithms.

Label/Method	Ref. [2]	Ref. [4]	Ref. [10]	Ref. [7]	Ref. [11]	Ref. [1]	Ref. [24]	Ref. [21]	Proposed
0	98.93	95.5	99.47	99.44	99.7	99.09	99.74	<b>99.97</b>	99.75
1	97.25	97.4	<b>99.78</b>	99.55	99.5	98.22	99.55	99.40	99.75
2	96.58	94.0	97.93	98.34	99.0	98.5	98.60	98.51	<b>99.65</b>
3	98.76	98.1	97.89	98.68	98.5	99.29	98.94	99.29	<b>99.70</b>
4	98.21	96.8	98.54	98.52	98.5	99.20	99.39	99.25	<b>99.80</b>
5	99.15	98.6	99.08	99.40	99.3	98.40	98.13	99.40	<b>99.79</b>
6	99.54	99.0	99.11	98.90	99.6	99.49	99.74	99.70	<b>99.80</b>
7	99.79	99.8	99.66	<b>100</b>	99.8	99.89	99.75	99.95	99.95
8	99.80	98.8	99.77	<b>100</b>	99.9	99.69	99.95	98.81	99.80
9	99.14	95.9	99.38	98.86	99.3	99.49	<b>99.94</b>	99.8	99.85

Table 13: Comparison of F score criteria between the proposed method and the state-of-the-art algorithms.

Label/Method	Ref. [2]	Ref. [4]	Ref. [10]	Ref. [7]	Ref. [11]	Ref. [1]	Ref. [24]	Ref. [21]	Proposed
0	98.3618	97.0252	99.3448	99.2446	99.5999	98.6430	99.1161	<b>99.8348</b>	99.7250
1	98.2643	97.5996	99.8350	99.5500	99.6996	98.8065	99.6998	99.5996	<b>99.8748</b>
2	97.7994	96.6246	98.3532	98.2199	98.5984	98.6498	98.7248	98.9778	<b>99.7000</b>
3	97.2059	94.8455	97.5236	98.1876	98.5000	99.0444	98.8700	98.8934	<b>99.7250</b>
4	98.5786	96.2974	98.5400	99.2545	98.9480	99.3996	99.3450	99.3000	<b>99.8000</b>
5	99.2499	97.1798	99.1999	99.4250	99.3999	98.6991	98.9580	99.4749	<b>99.7200</b>
6	99.1182	96.9588	99.1050	98.9250	99.3996	99.3949	99.6950	99.4996	<b>99.7750</b>
7	99.6949	<b>99.8999</b>	99.6800	99.4470	99.6999	99.7448	99.8000	99.6742	99.8749
8	99.8250	99.3964	99.8300	<b>100.0000</b>	99.9500	99.5448	99.9500	99.2778	99.8500
9	98.9697	97.5221	99.2699	99.4267	99.3999	99.3448	99.5686	99.4236	<b>99.8250</b>






handwritten digit					
True Label	0	0	5	5	5
Detected label	3	5	0	4	0

Figure 11: Samples of false recognition by the proposed network

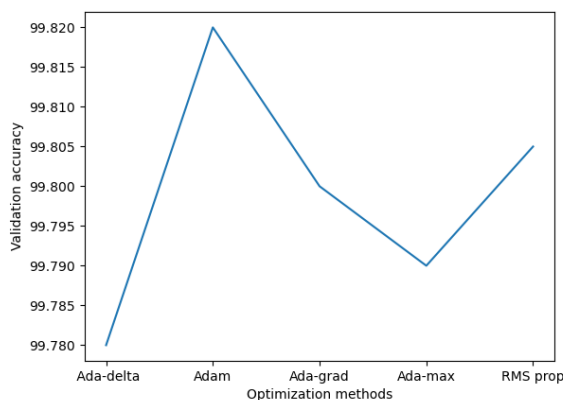


Figure 12: obtained maximum accuracies utilizing different optimization algorithm

## 5 Conclusion and Future Work

This paper introduces a novel architecture for Persian digit recognition that leverages dilated convolutional layers for efficient feature extraction. Subsequently, these feature maps are fed into a standard convolutional neural network to capture higher-level representations. A softmax layer is then used for digit classification. Experiments conducted on Google Colab using a Tesla T4 GPU over 500 epochs demonstrate the effectiveness of the proposed architecture, achieving 99.82% validation accuracy, 99.79% test accuracy, and 100% accuracy on the training set. The proposed network is distinguished by its simplicity, speed, and reduced implementation time compared to state-of-the-art methods due to its streamlined architecture. Future research could explore the application of this approach to car license plate detection or investigate further optimizations for computational efficiency.

## References

- [1] E. Al-wajih, and R. Ghazali, "Improving the accuracy for offline Arabic digit recognition using sliding window approach," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, pp. 1633–1644, 2020.
- [2] A. Alaei, U. Pal, and P. Nagabhushan, "Using modified contour features and SVM based classifier for the recognition of Persian/Arabic handwritten numerals," *Seventh International Conference on Advances in Pattern Recognition*, pp. 391-394, 2009.
- [3] M. J. Abdi, and H. Salimi, "Farsi handwriting recognition with mixture of RBF experts based on particle swarm optimization," *International Journal of Science and Computer and Mathematics*, vol. 2, pp. 129–136, 2010,

- [4] H. Salimi, and D. Giveki, "Farsi/Arabic handwritten digit recognition based on ensemble of SVD classifiers and reliable multi-phase PSO combination rule," *International Journal on Document Analysis and Recognition*, vol. 16, I. 4, pp 371–386, 2013 .
- [5] N. Loizou, and P. Richtárik, "Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods," *Computational Optimization and Applications* , vol. 77, pp. 653–710, 2020.
- [6] W. M. Pan, T.D. Bui, and C.Y. Suen, "Isolated handwritten Farsi numerals recognition using sparse and over-complete representations," *10th International Conference on Document Analysis and Recognition*, pp. 586–590, 2009.
- [7] M. J. Parseh and M. Meftahi, "A new combined feature extraction method for Persian handwritten digit recognition," *International Journal of Image and Graphics*, vol. 17, no. 02, p. 1750012, 2017.
- [8] J. Sadri, M. R. Yeganehzad, and J. Saghi, "A novel comprehensive database for offline Persian handwriting recognition," *Pattern Recognition*, vol. 60, pp. 378-393, 2016.
- [9] Y. Nanekaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, and 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*, pp. 1-30, 2020.
- [10] H. Sajedi, "Handwriting recognition of digits, signs, and numerical strings in Persian," *Computer and Electrical Engineering*. vol. 49, pp. 52–65, 2016.
- [11] G. A. Montazer, M. A. Soltanshahi, and D. Giveki, "Farsi/Arabic handwritten digit recognition using quantum neural networks and bag of visual words method," *Optical Memory and Neural Networks*, vol. 26, no. 2, pp. 117–128, 2017
- [12] S. Khorashadizadeh, and A. Latif, "Arabic/Farsi handwritten digit recognition using histogram of oriented gradient and chain code histogram," *International of Arab Journal of Information Technology*, vol. 13, no. 4, pp. 367–274, 2016.
- [13] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304, 2017.
- [14] V. Hajihashemi, M. M. A. Ameri, A. A. Gharahbagh, and A. Bastanfard, "A pattern recognition based Holographic Graph Neuron for Persian alphabet recognition," *International Conference on Machine Vision and Image Processing (MVIP)*, pp. 1-6, 2020.



- [15] M. Akhlaghi and V. Ghods, "Farsi handwritten phone number recognition using deep learning," *Applied Sciences*, vol. 2, no. 3, pp. 1-10, 2020.
- [16] F. Sarvaramini, A. Nasrollahzadeh, M. Soryani, "Persian handwritten character recognition using convolutional neural network" *Iranian Conference on Electrical Engineering (ICEE)*, pp. 1676–1680, 2018.
- [17] M. Parseh, M. Rahmanimanesh, and P. Keshavarzi, "Persian handwritten digit recognition using combination of convolutional neural network and support vector machine methods," *International Arab Journal of Information Technology*, vol. 17, no. 4, pp. 572-578, 2020.
- [18] E. Farahbakhsh, E. Kozegar and M. Soryani, "Improving persian digit recognition by combining data augmentation and AlexNet," *10th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 265–270. 2017.
- [19] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pages 1106–1114, 2012.
- [20] A. Bossaghzadeh, "Improving Persian digit recognition by combining deep neural networks and SVM and using PCA" *International Conference on Machine Vision and Image Processing (MVIP)*, pp. 1–5, 2020.
- [21] F. Haghighi, and H. Omranpour, "Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition," *Knowledge-Based Systems*, vol. 220, pp. 1–13, 2021.
- [22] H. Kusetogullari, A. Yavariabdi, J. Hall, and N Lavesson, "DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset," *Big Data Research*, vol. 23, pp. 1–12, 100182, 2021.
- [23] M. Bonyani, S. Jahangard and M. Daneshmand, "Persian handwritten digit, character and word recognition using deep learning," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 24, pp. 133–143, 2021.
- [24] V. M. Safarzadeh, and P. Jafarzadeh, "Offline persian handwriting recognition with CNN and RNN-CTC," *25th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1-10, 2020.
- [25] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: a real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935-1944, 2020.
- [26] R. Safdari, and M. S. Moin, "A hierarchical feature learning for isolated Farsi handwritten digit recognition using sparse autoencoder," *Artificial Intelligence and Robotics (IRANOPEN)*. 2016.

- [27] N. Ghanbari, "A review of research studies on the recognition of Farsi alphabetic and numeric characters in the last decade," *Fundamental Research in Electrical Engineering* pp. 173-184, Springer, Singapore, 2019.
- [28] A. Miri, and M. Khedmati, "Development of an Algorithm for Persian Handwriting Digits Recognition Based on MLP and PNN Classifiers and Using Cluster Centers," *Sharif Journal of Industrial Engineering and amp; Management*, 2023 in press.
- [29] H. Khosravi, and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition Letters*, vol. 28, I. 10, pp. 1133–1141, 2007.
- [30] A. El-Sawy, M. Loey, and H. EL-Bakry, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research*, vol. 5, pp. 11–19, 2017.
- [31] C. Szegedy, w. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.
- [32] K. He, X. Zhang, Sh. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 770–778, 2016.
- [33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu and T. S. Huang, "Generative image inpainting with contextual attention," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5505-5514, Salt Lake City, UT, USA, 2018.