Persian Handwritten Digit Recognition Using Combination of Convolutional Neural Network and Support Vector Machine Methods

Mohammad Parseh, Mohammad Rahmanimanesh, and Parviz Keshavarzi
Faculty of Electrical and Computer Engineering, Semnan University, Iran

Abstract: *Persian handwritten digit recognition is one of the important topics of image processing which significantly considered by researchers due to its many applications. The most important challenges in Persian handwritten digit recognition is the existence of various patterns in Persian digit writing that makes the feature extraction step to be more complicated.Since the handcraft feature extraction methods are complicated processes and their performance level are not stable, most of the recent studies have concentrated on proposing a suitable method for automatic feature extraction. In this paper, an automatic method based on machine learning is proposed for high-level feature extraction from Persian digit images by using Convolutional Neural Network (CNN). After that, a non-linear multi-class Support Vector Machine (SVM) classifier is used for data classification instead of fully connected layer in final layer of CNN. The proposed method has been applied to HODA dataset and obtained 99.56% of recognition rate. Experimental results are comparable with previous state-of-the-art methods.*

Keywords: *Handwritten Digit Recognition, Convolutional Neural Network, Support Vector Machine.*

Received January 1, 2019; accepted November 11, 2019
https://doi.org/10.34028/iajit/17/4/16

1. Introduction

Optical Character Recognition (OCR) is one of the attractive topics of Artificial Intelligence [3, 6, 15, 23, 24]. In this topic, handwritten digit recognition has a significant importance in different languages due to its various usages such as automatic check reading in banks, automatic reading of forms filled by human, etc. The handwritten digit recognition is often consideredas a multi-class classification task which aims to classify the images into different digit classes (i.e., the number of classes is equal to the number of digits).

Support Vector Machine (SVM) [2, 17, 18] is a popular classification method which iswidely used for handwritten digit recognition. In this paper, we address the Persian handwritten digit recognition problem as a multi-class SVM-based classification task with 10 different classes. Besides, feature extraction is one of the challenging phases in this classification task, because the writing styles of Persian handwritten digitsare very diverse. The previous works that used SVM classifier, first extracted the appropriate features of data by using handcraft feature extraction methods, and then classified the obtained feature vectors by using SVM classifier [17, 14]. Handcraft feature extraction methods are complicated, and time-consuming, often producing non-stable results [14]. Since Convolutional Neural Network (CNNs) have achieved remarkable results in feature extraction in OCR tasks in the last few years, they are good alternatives to handcraft feature extraction method.

2. Related Works

Recent researches in the handwritten digit recognition taskaim to extract and select the appropriate features from images in such a way that the classes of the digits can be separated from each other correctly [1, 4,7,16, 20]. Parseh and Meftahi [18] managed to accomplish %99.07 of recognition rate for Persian handwritten. The most important handcrafted features used in this study are hole size, writing direction and number of endpoints in each area. They used PCA to reduce the dimensions of the feature vector, and used SVM for data classification. To recognize the Persian handwritten digits by SVM classifier, Nooraliei [17] extracted features like zoning and histogram projection on 8000 samples of HODA dataset and obtained 97.83% of recognition rate. Rashnoudi in [21, 22] used Discrete Fourier Transform (DFT) coefficients as features for classifying 6000 data samples. In [2], a combination of a novel feature set is introduced based on the transition information in horizontal and vertical directions and the famous chain of Freeman. The authors used SVM method for data classification and obtained 98.55% of recognition accuracy on HODA dataset. Hajizade *et al*. [9] proposed a novel non-linear learning technique based on Locally Linear Embedding (LLE). After that, by applying Linear Discrimination Analysis (LDA) on the images of Persian handwritten characters and using KNN classifier, they classified the data samples. Zamani *et al*. [26] obtained %99.03 of recognition rate on HODA dataset by combining CNN and random forest method. Khorashadizadeh and Latif [11] have been constructed a feature set by combination of four directional Chain Code Histogram (CCH) and Histogram of Oriented Gradient (HOG). They are used SVM for classification and achieved %99.31of classification rate on HODA dataset.

3. Proposed Method

Due to the vast variety of writing patterns in Persian digits, the feature extraction step is very important. Handcraft feature extraction methods are very complicated and have low and non-stable performance [2, 17, 21, 22]. Therefore, in this study, a CNN is used to extract the appropriate features from images because it shows high performance in extracting features from images [8, 13, 20].

The block diagram of the proposed Persian handwritten digit recognition method is shown in Figure 1. In themethod, after feature extraction by using CNN, two types of classifiers are used for classifying Persian handwritten digits. The first one is fully connected layer which is placed as a classifier at the end of the CNN and is used to fine-tune the kernel’s weight in the training phase. The second one is a non-linear SVM which is used as a classifier after extracting the high-level features by using convolutional layer. We will evaluate the performance of each of these classifiers and compare their results in section 4.

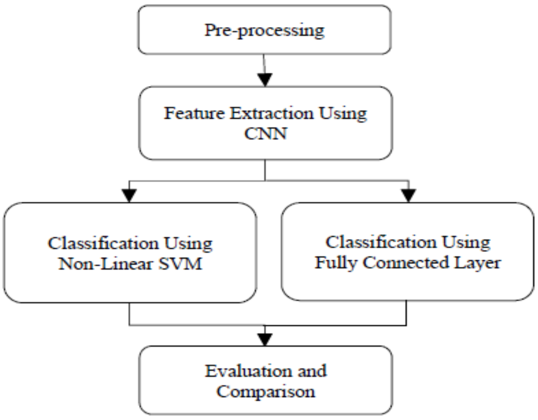

Pre-processing
Feature Extraction Using CNN
Classification Using Non-Linear SVM
Classification Using Fully Connected Layer
Evaluation and Comparison

Figure 1. The block diagram of our proposed method.

3.1. Pre-Processing

We use HODA Persian handwritten digit dataset to evaluate the performance of the proposed method. Since the dimensions of the images are different, we must unify their dimensions for using them in the input layer of CNN. In the pre-processing step, the images are converted to binary matrices in such a way that the foreground pixels have set to 1 and the background pixels have set to 0. After binarizing the images, we resize the dimensions of the images to 40*40 and the existing patterns have been placed in the centre of the images. Figure 2 shows some samples of HODA dataset after pre-processing.

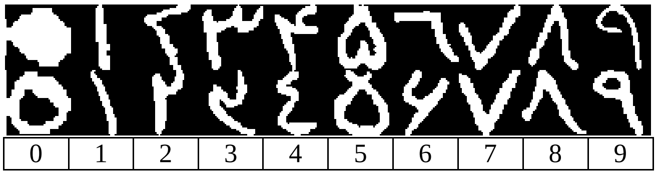


Figure 2. Some samples of HODA dataset after pre-processing.

3.2. Convolutional Neural Network (CNN) Structure

In recent years, CNN has been widely used in image processing applications and has proved its position in this field due to its amazing results [5, 13]. Especially, CNNis growing increasingly recently for extracting high-level features from images [8].This type of network is not only a deep neural network with a large number of hidden layers and neurons, but also is a network which simulates the vision process of the human brain for recognizing the images.

Recognizing handwritten digits in images can be considered as an image classification task that classifies the images to different digit classes. Before convolutional neural network becomes widespread among researchers, feature extraction was carried out by experts apart from machine learning process. In convolutional neural networks, feature extraction is not done manually, but automatically in a machine learning process by setting the kernel’s weight. Figure 3 shows the proposed network structure for training phase.

In the proposed network structure, the parameters of the network such as depth, number of neurons in each layer, and etc., are determined in such a way that keep a trade-off between the training time and the recognition rate. As it can be seen in Figure 3, this network contains an input layer at the start and an output layer at the end of the network. The input layer has 40*40*1 neurons (the same as the dimensions of the input images). Commonly, CNNs perform data classification task by using a fully connected layer as their final layer (like a MLP neural network). However, in this paper we use the fully connected layer to fine-tune the weights of the kernels in the training phase. Additionally, in final step of the training phase, we also train the model by using a non-linear multi-class SVM on the output of last layer of the feature extractor module.

Feature extractor module, which is placed between the input layer and the fully connected layer, extracts the high-level features from the input images. It is clear that the upper layers in CNNs extract the higher level of features [19]. As it is shown in Figure 3, the feature extractor module consists of six blocks in which each block continues with a max pooling layer.

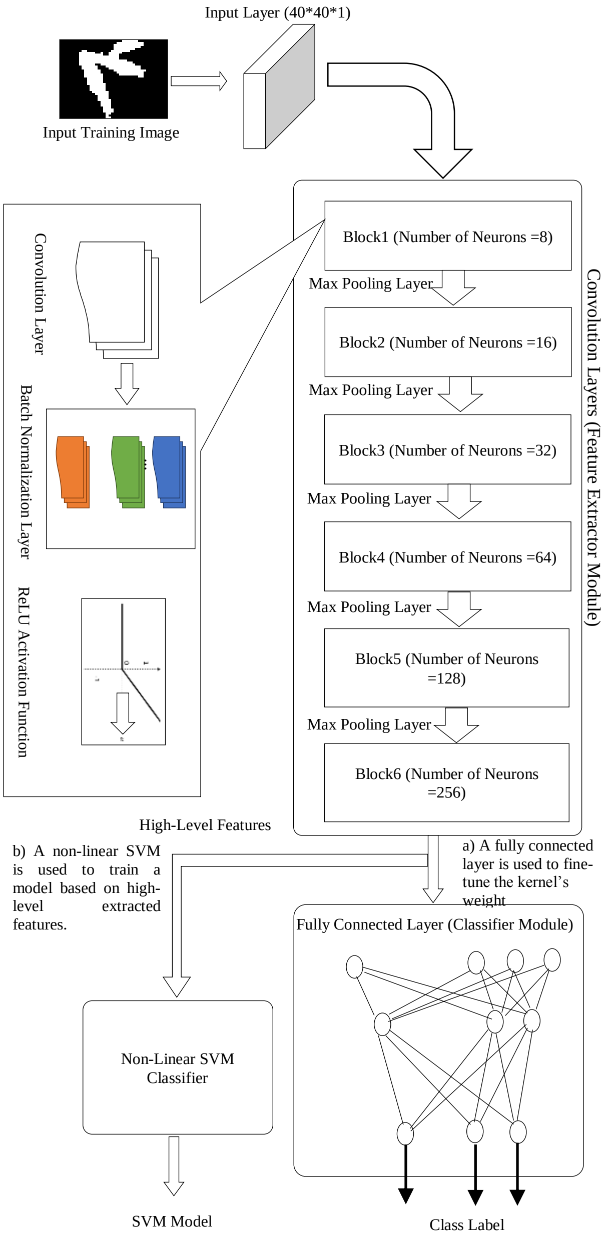
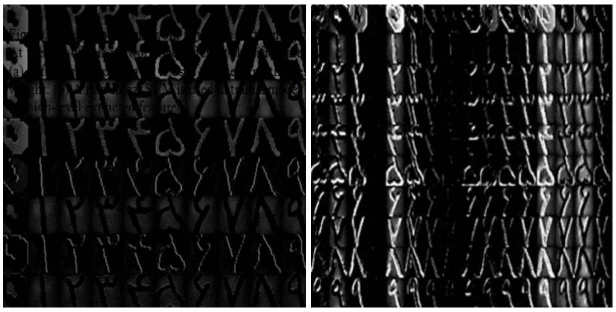

Input Layer (40*40*1)
Input Training Image
Convolution Layer
Batch Normalization Layer
ReLU Activation Function
Block1 (Number of Neurons =8)
Max Pooling Layer
Block2 (Number of Neurons =16)
Max Pooling Layer
Block3 (Number of Neurons =32)
Max Pooling Layer
Block4 (Number of Neurons =64)
Max Pooling Layer
Block5 (Number of Neurons =128)
Max Pooling Layer
Block6 (Number of Neurons =256)
Convolution Layers (Feature Extractor Module)
High-Level Features
b) A non-linear SVM is used to train a model based on high-level extracted features.
a) A fully connected layer is used to fine-tune the kernel's weight
Fully Connected Layer (Classifier Module)
Non-Linear SVM Classifier
SVM Model
Class Label

Figure 3. Our proposed network structure for training phase. At the end of this network two different classifiers are used.

Pooling operation helps the network to focus on local features and to speed up the training process by decreasing the size of the feature maps. There are many pooling operations such as max pooling, average pooling and etc. Since we used binary images, the max pooling was better choice for pooling operation than others.

In Figure 3, each block contains three layers that are as follows: 1- Convolutional layer that uses a 2D convolution operator along with a determined number of kernels. Number of kernels used in each convolutional layer are 8, 16, 32, 64, 128, and 256, respectively. Increasing the number of kernels along the network's layers leads to obtaining higher level features from the input images, but on the other side, increasing the computational complexity and time burden. Figure 4 shows the extracted features from some sample images by the first and second convolutional layers.2- Batch normalization layer removes correlation between data by normalizing the batches of data independently. More information about this layer have been presented in section 3.2.1. 3- Applying the activation function to the output of batch normalization layer. In the proposed network, ReLU function is used as activation function. This function is linear for positive real numbers and is more practical in this application than the other activation functions such as sigmoid and tangent hyperbolic.



a) Each column consists of feature maps of a sample image of digits that is output of first convolutional layer. In this layer,the number of kernels is 8 and the size of each kernel is 3.

b) Each row consists of feature maps of a sample image of digits that is output of second convolutional layer. In this layer,the number of kernels is 16 and the size of each kernel is 3.

Figure 4. Feature maps from CONV1 and CONV2 layers.

In the testing phase, we first extract the high-level features from the test images by using the feature extractor module and then classify it by the trained model of SVM. Figure 5 shows the proposed network structure in the testing phase.

3.2.1. Batch Normalization

Training a deep neural network is complicated and time-consuming process because a change in the parameters of one layer changes the distribution of inputs of next layers. We know this phenomenon as internal covariance shift, and we normalize the inputs of each layer to overcome that. Batch normalization (BN) allows us to use higher learning rate and to be less careful for the initial valuing. Since the whitening of the input of each layer is expensive, in this method each scalar feature is normalized independently instead of whitening of input and output features of each layer jointly [10]. This is done by transforming the data into new scale with a mean of 0 and standard deviation of 1 in all feature dimensions. For a layer with d-dimensional input$X = \left(x^{(1)}, x^{(2)}, \dots, x^{(d)}\right)$, each dimension will be normalized by:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right]}} \qquad (1)$$

Where expectation ($E\left[x^{(k)}\right]$) and variance ($Var\left[x^{(k)}\right]$) will be computed on the whole training dataset. As it is shown in [10], this method leads to convergence speed up, even when the features are not uncorrelated.

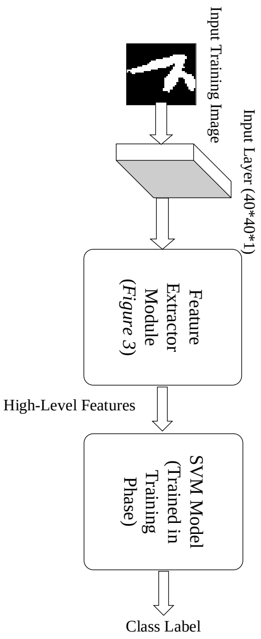

Input Training Image
Input Layer (40*40*1)
Feature Extractor Module (Figure 3)
High-Level Features
SVM Model (Trained in Training Phase)
Class Label

Figure 5. The proposed network structure for testing phase.

It should be noted that normalizing the input of each layer might change what a layer would represent. For example, normalizing the inputs of a sigmoid activation function might restrict them to a linear regime. To address this, we must make sure that the transformation inserted in the network can represent the identity transform [10]. In doing it, we define parameters $\beta^{(k)}$ and $y^{(k)}$ for $x^{(k)}$ which shifts and scales the normalized values respectively:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \qquad (2)$$

These parameters will be learned with the main parameters of the model during the training phase.

Consider the mini-batch of β with size of m. Since the normalization is done independently on each activation, we can focus on the particular activation $x^{(k)}$ and omit k for simplification. m values of this activation are placed in mini-batch β as:

$$\beta = \{x_{1\,\dots\,m}\} \qquad (3)$$

Assume that $\hat{x}_{1\,\dots\,m}$ are normalized values and $y_{1}...m$ are the linear transformation of the values. We define batch normalizing transform as follows:

$$BN_{\gamma,\beta} : x_{1\dots m} \rightarrow y_{1\dots m} \qquad (4)$$

Algorithm1 presents flowchart of BN Transform algorithm, where ε is a constant value which is added to mini-batch variance for numerical stability.

3.3. Support Vector Machine (SVM) Classifier

SVM is one of the most popular classifiers in the past two decades which has shown good performance in the most applications such as image processing. In non-linear SVM, data is transformed to a new space (hybrid space) by using a kernel function in such a way that the transformed data are separable in new space. Then, a model will be obtained by training the binary separators and combining them so that the data can be classified to several different classes. Equation (5) shows a non-linear SVM classifier [25]:

Algorithm 1. Batch Normalizing Transform [10]. γ, β are Parameters to be learned.

```
# Input: Values of x over a mini-batch: ${x_{1\,\dots\,m}}$;
            Parameters to be learned: γ, β
$y_i = BN_{\gamma,\beta}(\{x_{1\,\dots\,m}\})$
{
    #mini-batch mean
for(i=1 to m)
                    $sum = sum + x_i$
                    $mean = sum/m$
    #mini-batch covariance
for(i=1 to m)
                    $dev = dev + (x_i - \mu_\beta)^2$
                    $var = dev/m$
    # normalization
for(i=1 to m)
                    $\hat{x}_i = (x_i - mean)/\sqrt{var + \varepsilon}$
    # scale and shift
for(i=1 to m)
                    $y_i \leftarrow \gamma\hat{x}_i + \beta$

return$\{y_{1\,\dots\,m}\}$
}
```

$$f(x) = Sign\left(\sum_{SupportVectors} y_i \alpha_i K(x_i, x_j) + b\right) \qquad (5)$$

Where $K(x_i,x_j)$ is kernel function and b is bias. The most common kernel functions are: RBF (Gaussian), sigmoid and polynomial.

4. Experimental Results

We used HODA dataset to evaluate our proposed method, which is the most common dataset of Persian handwritten digits [12]. This dataset contains 102352 samples totally with resolution of 200dpi where 60000 of them are training samples (6000 samples for each class) and 20000 of them are testing samples (2000 samples for each class). The experiments have been carried out by Matlab 2017b software on Intel Core i5 CPU with Nvidia GEFORCE GTX1070 graphical processor.

As it is mentioned in section 3, CNN has been used for extracting the appropriate features from the images. Additionally, we used a non-linear SVM classifier along withthe fully connected layer in order to improve the classification performance. The results of both methods are shown in Table 1.

Table 1.Comparingthe classifiers with different number of layers in CNN.

	Percentage of recognition (%)	
	Number of Convolutional layers = 5	Number of Convolutional layers = 6
CNN + Fully connected layer for classification	99.33	99.40
CNN + Non-linear SVM for classification	99.42	99.56

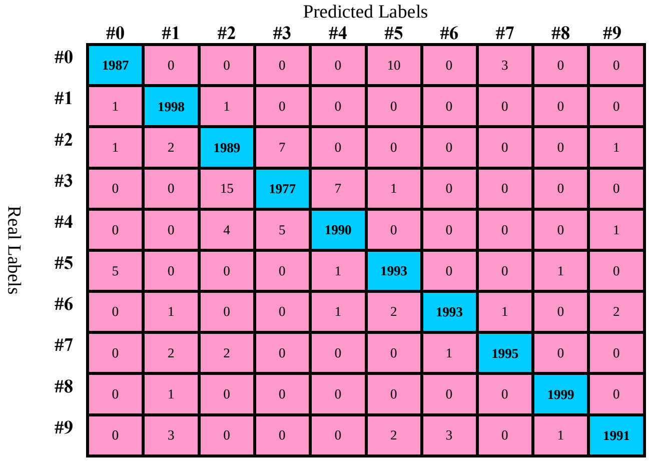
The experimental results in Table 1 show that using a non-linear SVM classifier at the end of CNN increases the performance of the classification task as compared to using fully connected layer. Table 2 shows that our proposed method outperforms the best recent methods. Additionally, by using the batch normalization technique in each convolutional layer, the training time decreases as much as possible. Table 3 shows the precision, recall, F-measure and recognition rate of each digit individually.

Table 2. Comparing the proposed method with previous methods.

Method	Accuracy of Test (%)
Sajedi *et al.* [22], SVM with polynomial kernel (HODA dataset)	99.07
Parseh and Meftahi [18], Novel hand-craft feature extraction method (HODA dataset)	99.07
Boukharouba *et al*. [23], SVM with novel features (HODA dataset)	98.55
Zamani *et al.* [26], CNN, LeNet Network (HODA dataset)	99.03
Our Proposed Method, Combination of CNN and SVM (HODA dataset)	99.56

Table 3. Precision, recall, F-measure and accuracy of each class.

Digits	Precision (%)	Recall (%)	F-measure	Accuracy (%)
#0	99.65	99.35	99.49	99.35
#1	99.55	99.90	99.72	99.90
#2	98.91	99.45	99.17	99.45
#3	99.40	98.85	99.12	99.85
#4	99.55	99.50	99.52	99.50
#5	99.25	99.65	99.45	99.65
#6	99.80	99.65	99.72	99.65
#7	99.80	99.75	99.77	99.75
#8	99.90	99.95	99.92	99.95
#9	99.80	99.55	99.67	99.55



Predicted Labels

Real Labels

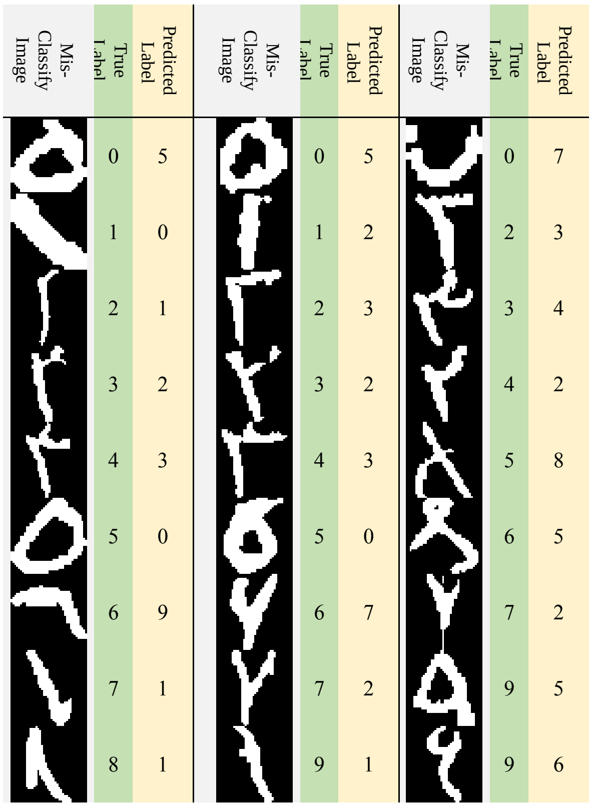
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9
#0	1987	0	0	0	0	10	0	3	0	0
#1	1	1998	1	0	0	0	0	0	0	0
#2	1	2	1989	7	0	0	0	0	0	1
#3	0	0	15	1977	7	1	0	0	0	0
#4	0	0	4	5	1990	0	0	0	0	1
#5	5	0	0	0	1	1993	0	0	1	0
#6	0	1	0	0	1	2	1993	1	0	2
#7	0	2	2	0	0	0	1	1995	0	0
#8	0	1	0	0	0	0	0	0	1999	0
#9	0	3	0	0	0	2	3	0	1	1991

Figure 6. Number of correctly and incorrectly classified samples of each digit.

Figure 6 shows that how many samples of each digit are recognized correctly and incorrectly at the testing phase. As it is shown in Figure 6, the most misclassified samples occurred in digit “3”, where the most of them were recognized incorrectly as digit “2”. This misclassification emanates from great similarity in the writing pattern of digits “2” and “3”. Figure 7 shows some samples of digits which are classified incorrectly because they are very similar to other digits. As it is shown in Figure 7, most of the misclassified samples are not recognizable even by human eyes.

5. Conclusions

A combined method for Persian handwritten digit recognition has been proposed in this study. In the method, we extracted high-level features of digits by a convolutional neural network at first and then we classified the feature vectors by a fully connected layer as a classifier. Additionally, we used a non-linear SVM classifier instead of fully connected layer which caused improvement in the performance of classification. Also, the experimental results show that the proposed method outperformsthe previous works. Moreover, the structure of CNN in this method is designed in such a way that the training time becomes acceptable. As the future work, we aim to apply a relatively similar method for recognizing Persian handwritten characters. Of course, more difficulty in writing patterns and also more number of classes will be the most important challenges in that work.



Mis-Classify Image	True Label	Predicted Label	Mis-Classify Image	True Label	Predicted Label	Mis-Classify Image	True Label	Predicted Label
	0	5		0	5		0	7
	1	0		1	2		2	3
	2	1		2	3		3	4
	3	2		3	2		4	2
	4	3		4	3		5	8
	5	0		5	0		6	5
	6	9		6	7		7	2
	7	1		7	2		9	5
	8	1		9	1		9	6

Figure 7. Some misclassified samples of digits which are very similar to other digits.

References

[1] Alizadehashraf B. and Roohi S., "Persian Handwritten Character Recognition Using Convolutional Neural Network," *in Proceedings of 10th Iranian Conference on Machine Vision and Image Processing*, Isfahan, pp. 247-251, 2017.

[2] Boukharouba A. and Bennia A., "Novel Feature Extraction Technique for The Recognition of Handwritten Digits," *Applied Computing and Informatics*, vol. 13, no. 1, pp. 19-26, 8 2017.

[3] Broumandnia A., Shanbehzadeh J., and Varnoosfaderani M., "Persian/Arabic Handwritten Word Recognition Using M-Band Packet Wavelet Transform," *Image and Vision Computing*, vol. 26, no. 6, pp. 829-842, 2008.

[4] Chherawala Y. and Cheriet M., "Arabic Word Descriptor for Handwritten Word Indexing and Lexicon Reduction," *Pattern Recognition*, vol. 47, no. 10, pp. 3477-3486, 2014.

[5] Chollet F., *Deep Learning With Python*, Manning Publications Co, 2018.

[6] Dehghan M., Faez K., Ahmadi M., and Shridhar M., "Handwritten Farsi (Arabic) Word Recognition: A Holistic Approach Using Discrete HMM," *Pattern Recognition*, vol. 34, no. 5, pp. 1057-1065, 2001.

[7] Farahbakhsh E., Kozegar E., and Soryani M., "Improving Persian Digit Recognition By Combining Data Augmentation and Alexnet," *in Proceedings of 10th Iranian Conference on Machine Vision and Image Processing*, Isfahan, pp. 265-270, 2017.

[8] Goodfellow I., Bengio Y., and Courville A., *Deep Learning*, MIT Press, 2016.

[9] Hajizadeh R., Aghagolzadeh A., and Ezoji M., "Manifold based Persian Digit Recognition Using the modified Locally Linear Embedding and Linear Discriminative Analysis," *in Proceedings of 2nd International Conference on Knowledge-Based Engineering and Innovation*, Tehran, pp. 614-618, 2015.

[10] Ioffe S. and Szegedy C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *in Proceedings of the 32nd International Conference on Machine Learning*, Lille, 448-456, 2015.

[11] Khorashadizadeh S. and Latif A., "Arabic/Farsi Handwritten Digit Recognition usin Histogra of Oriented Gradient and Chain Code Histogram," *The International Arab Journal of Information Technology*, vol. 13, no. 4, pp. 367-374, 19 2016.

[12] Khosravi H. and Kabir E., "Introducing a Very Large Dataset of Handwritten Farsi Digits and A Study on Their Varieties," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1133-1141, 2007.

[13] Kim P., "Matlab DEEP LEARNING," *With Machine Learning, Neural Networks and Artificial Intelligence*, vol. 130, 2017.

[14] Liang H., Sun X., Sun Y., and Gao Y., "Text Feature Extraction Based on Deep Learning: A Review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 1-12, 2017.

[15] Mozaffari S., Faez K., Märgner V., and El-Abed H., "Lexicon Reduction Using Dots for Off-Line Farsi/Arabic Handwritten Word Recognition," *Pattern Recognition Letters*, vol. 29, no. 6, pp. 724-734, 2008.

[16] Naz S., Umar A., Ahmad R., Siddiqi I., Ahmed S. B., Razzak M., and Shafait F., "Urdu Nastaliq Recognition Using Convolutional-Recursive Deep Learning," *Neurocomputing*, vol. 243, pp. 80-87, 13 2017.

[17] Nooraliei A., "Persian Handwritten Digits Recognition By Using Zoning and Histogram Projection," *in Proceedings 3rd Joint Conference of AI and Robotics and 5th RoboCup Iran Open International Symposium*, Tehran, pp. 1-5, 2013.

[18] Parseh M. and Meftahi M., "A New Combined Feature Extraction Method for Persian Handwritten Digit Recognition," *International Journal of Image and Graphics*, vol. 17, no. 02, pp. 1750012, 2017.

[19] Patterson J. and Gibson A., *Deep learning: A Practitioner's Approach*. O'Reilly Media, Inc, 2017.

[20] Pourmohammad A. and Ahadi S., "Using Single-Layer Neural Network for Recognition of Isolated Handwritten Persian Digits," *in Proceedings of 7th International Conference on Information, Communications and Signal Processing*, Macau, pp. 1-4, 2009.

[21] Rashnodi O., Sajedi H., and Abadeh M., "Persian Handwritten Digit Recognition Using Support Vector Machines," *International Journal of Computer Applications*, vol. 29, no. 12, pp. 1-6, 2011.

[22] Rashnodi O., Sajedi H., and Abadeh M., "Using Box Approach in Persian Handwritten Digits Recognition," *International Journal of Computer Applications*, vol. 32, no. 3, pp. 1-8, 2011.

[23] Razzak M., Anwar F., Husain S., Belaid A., and Sher M., "HMM And Fuzzy Logic: A Hybrid Approach for Online Urdu Script-Based Languages' Character Recognition," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 914-923, 2010.

[24] Saeed K. and Albakoor M., "Region Growing Based Segmentation Algorithm for Typewritten and Handwritten Text Recognition," *Applied Soft Computing*, vol. 9, no. 2, pp. 608-617, 2009.

[25] Vapnik V., *The Nature of Statistical Learning Theory*, Springer science and Business Media, 2013.

[26] Zamani Y., Souri Y., Rashidi H., and Kasaei S., "Persian Handwritten Digit Recognition By Random Forest and Convolutional Neural Networks," *in Proceedings of 9th Iranian Conference on Machine Vision and Image Processing*, Tehran, pp. 37-40, 2015.

Mohammad Parseh received his first degree in Software Engineering from Shahid Chamran University of Ahwaz in 2008 and his MSc degree in Artificial Intelligence from Tabriz University in 2012. Currently, he is PhD candidate in Semnan University, Iran. His interested topics are handwritten character recognition, scene understanding, visual object tracking, deep learning, big data analysis and machine learning.

Mohammad Rahmanimanesh received his MS and PhD both from the TarbiatModares University, Tehran, Iran, and BS from the Sharif University ofTechnology, Tehran, Iran, all in Computer Engineering. He is currently an Assistant Professor at Semnan University, Semnan, Iran. He is a member of IEEE and his research interests include network security, fuzzy systems, softcomputing, and data mining.

Parviz Keshavarzi received the M.S. degree in electronic engineering from Tehran University, Tehran, Iran, in 1988 and the Ph.D. degree in electrical engineering from the University of Manchester, Manchester, U.K., in 1999. He is currently an Associate Professor with Semnan University, Semnan, Iran. His research interests include neuromorphics and nanoelectronics.