Persian Handwritten Digit Recognition by Random Forest and Convolutional Neural Networks

Yasin Zamani, Yaser Souri, Hossein Rashidi and Shohreh Kasaei, *IEEE, Senior Member*
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
Emails: {yzamani, ysouri, hrashidi}@ce.sharif.edu and skasaei@sharif.edu

***Abstract*—Persian handwritten digit recognition has attracted some interests in the research community by introduction of large *Hoda* dataset. In this paper, the well-known *random forest* (RF) and *convolutional neural network* (CNN) algorithms are investigated for Persian handwritten digit recognition on the Hoda dataset. Using the Hoda dataset as a standard testbed, we have performed some experiments with different preprocessing steps, feature types, and baselines. It is then shown that RFs and CNNs perform competitively with the state-of-the-art methods on this dataset, while CNNs being the fastest if appropriate hardware is available.**

Keywords*—Machine learning, Random forest, Convolutional neural network, Handwritten digit recognition, Persian digits, Hoda dataset.

I. Introduction

The *optical character recognition* (OCR) has been one of the main interests of computer vision researchers. Also, lots of work has been done on recognition of Persian/Arabic handwritten letters and digits [1]. In the following, some papers which have used the Hoda dataset for their evaluation are listed (the accuracy of each method is mentioned in parentheses).

Parvin *et al.* [2] proposed an ensemble classifier (like decision tree) and optimized that with *genetic algorithm* (GA) (97.12%). In [3], authors create a GA-based method for constructing a neural network ensemble using a weighted vote-based classifier selection approach (98.27%). In [4] an ensemble classifier includes a base multi-class *multilayer perceptron* (MLP) and some binary classifiers which selected subject to confusion matrix are created and then used a GA to optimize weighted votes of this ensemble (98.89%). Authors in [5] proposed a new feature extraction method based on modified contour chain code and then used a *support vector machine* (SVM) to classify the data (98.71%). Hamidi and Borji in [6] created new features by modified *hierarchical model and x* (HMAX) feature extraction and then used a SVM to classify the data. Alaei *et al.* (99%) in [7] modified their last approach in [5] and used that with a modified transition features for distinguishing between classes which have a big confusion (99.02%). Authors in [8] combined the decisions of multiple classifiers and then used the SVD classifier for classifying the data (training and test sets are selected randomly, 97.02%).

In the late 80s Y. Lecun *et al.* showed that a typical structure of neural networks with multiple layers of convolution are well suited for the task of hand written digit recognition [9]. Years after CNNs are used for a wide array of tasks in computer vision [10]. R. Caruana *et al.* showed that RF classifiers and their variants can achieve the best performance on large number of datasets among many other classifiers [11]. In fact, Random forest [12] is an ensemble classifier which contains a large number of decision trees and its output decision is equal to the mode of singular trees decisions. RF algorithm was designed by Leo Breiman and Adele Culter and the RFs is its commercial mark. These results motivated us to try this kind of models for our supervised classification task of Persian handwritten recognition. Our experimental results indeed showed that both RF and CNNs are appropriate for this task. In fact, CNNs *learn* multiple layers of filters (feature extractors) from the raw data that are demonstrated to work better than any other engineered feature extractors.

The rest of this paper is organized as follows. In Section II the proposed method is introduced and justified. Experimental results and the related analysis are given in Section III. Finally, Section IV concludes the paper.

II. Proposed Method

In this section, the experiments that verify the power of RF for the task of Persian digit recognition are presented. The RF while being a simple and fast method, achieves comparable performance to the state-of-the-art methods in Persian digit recognition task. In our experiments, the Hoda dataset [13] is used; which is the most used and the largest Persian digit dataset to the best of our knowledge. This dataset consists of 60,000 training and 20,000 test image files each containing a single Persian digit. In all experiments, separate train and test sets of the Hoda dataset are used.

A. Preprocessing Stage

Table I shows that the training data of the Hoda dataset consists images with variant dimensions (each dimension can vary from 3 to 62 pixels). A statistical analysis which is performed on the training data, demonstrates that data in this set has an average dimension of $20(\pm 7)\times 29(\pm 8)$ pixels (details of this analysis for different digits is reported in Table I). Therefore, for omitting the effect of training image's size, all training data images are transformed to 32×32 pixels as a preprocessing step. Two different kinds of preprocessing methods are utilized. These are described below.

1) Scaling Stage: In this preprocessing stage, each image is scaled from its original dimension to 32×32 pixels by using the cubic interpolation. . For preserving more details, the cubic interpolation is used. While being simpler, this method has some side effects. For example, some samples of the Hoda dataset (containing digit 1) are very narrow (4×21) and

TABLE I. AVERAGE OF WIDTH AND HEIGHT OF HODA TRAINING DATA FOR EACH DIGIT LABELS.

Label	Dimensions (width × height)
0	13(±3) × 12(±3) px
1	09(±3) × 30(±5) px
2	17(±3) × 32(±5) px
3	24(±4) × 32(±5) px
4	22(±4) × 33(±6) px
5	25(±4) × 27(±5) px
6	21(±4) × 32(±5) px
7	24(±5) × 28(±5) px
8	24(±4) × 28(±5) px
9	21(±4) × 33(±5) px
Overall	**20(±7) × 29(±8) px**

scaling them to 32×32 will result in a huge distortion to the aspect ratio of the images of samples. But, since this scaling preprocess is also used in the test phase, this preprocessing method performs better when used in our RF pipeline.

2) Padding Stage: In this preprocessing stage, it is meant to maintain the aspect ratio of digits. First, the samples are grouped to a large (having one dimension larger than 32) and a small (both of its dimensions are smaller than 32) subsets. For the large subset, firstly each image is placed at the center, to the smallest square containing it with the background color of white color or intensity of 255. Next, each produced image is scaled down (by using the cubic interpolation) to the fixed dimension of 32×32 pixels. For the small subset, the image is placed at the center of a 32×32 image and the free space is padded with the intensity of 255 (white).

B. Feature Extraction Stage

After the preprocessing stage, different features have to be calculated to be used by RF. Two different kinds of features have been utilized. These features are described below.

1) Block Features: For block features, initially the preprocessed samples have been transformed to a binary image with intensity threshold of 128. Then, it is represented as an array of 4×4 pixels sub-region (starting from top left corner, see Figure 1). Then, the number of black pixels in each region (between 0 to 16) is a related feature in the feature vector. As a result, each sample is represented by a 64×1 feature vector. For example, in Figure 1, the first item in feature vector is 5.

2) HOG Features: As a replacement for block features, we have also used the famous *histogram of oriented gradient* (HOG) features [14]. These features are shown to perform well for rigid object recognition [14]. They encode appearance and shape of images. As such, the HOG descriptor was particularly well suited for this purpose. Nine gradient orientations with blocks of 8×8 pixels and cells of 2×2 blocks were used. Therefore, the final feature vector for a 32×32 pixel image has 324 dimensions.

C. Performance Evaluation Stage

As the performance evaluation criteria, the widely used *accuracy* measure is chosen. For the sake of completeness, the accuracy is

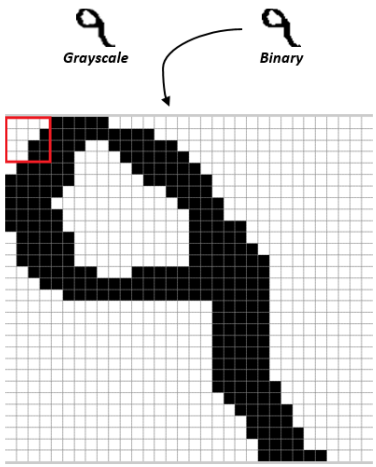

Fig. 1. Some image samples of Persian handwritten digits.

$$\frac{1}{20,000}\sum_{i=1}^{20,000}\mathbb{I}(y_i = f(x_i)) \quad (1)$$

where y_i is the true label for each testing image, x_i is the preprocessed testing image, $f(.)$ is the learned classifier, and $\mathbb{I}$ is the indicator function. Obviously, the error rate (which is used for performance evaluation of MNIST [2]) can easily be converted to accuracy ($errorrate = 1 - accuracy$).

The time that is consumed during learning and prediction for the whole dataset is also reported . To put things in context, a single machine with a quad core Intel core i7 4GHz CPU is used . Also, the CNN on one NVIDIA GTX 780 Ti 6GB GPU was trained.

III. EXPERIMENTAL RESULS

To show the efficiency of the proposed method, experiments with some baselines were conducted. For all these baselines the padding preprocessing have been used. Please note that in all baselines the raw pixel values after preprocessing as feature vectors are used. As such, each dataset image (after preprocessing) is transformed into a 1024 (32 × 32) dimensional feature vector, where each dimension is an integer between 0 and 255.

A. k-Nearest Neighbor Classifier

The *K-nearest neighbor* (KNN) classifier is a non-parametric model for classification that has a large test time cost. It is contrary to the RF which is very fast by design during the test time. But, still the nearest neighbor classifier is a very good testing method for a dataset. For 1-NN classification, the obtained accuracy is *96.19%* and, based on our experiments, increasing the number of k decreases the performance while even values for k results in a small degrade of accuracy. This small degrade is the result of equal number of votes for different classes from neighbors, which needs a tie breaker. A full run of the algorithm including the training and testing on the full dataset takes *18 minutes* on our machine.

B. Support Vector Machine Classifier

Another obvious choice for baseline is the SVM . A linear kernel with C value set to 1 was used. The accuracy measure was *89.91%*, while the time consumed for training and testing was *4 minutes and 25 seconds*, respectively. Our results indicated that for block features, the choice of preprocessing makes a difference. With block features, scaling preprocessing performs slightly (∼0.8%) better than padding preprocessing. For HOG features, in small ensemble sizes (below 8), the choice of padding preprocessing results in a small performance improvement. But, for larger ensembles the preprocessing does not have noticeable effect on the performance. It can be concluded that HOG features, in general, are better suited for the task of digit recognition. Also, the performance improves as the ensemble size increases. This improvement saturates at *32 trees*. Increasing the ensemble size to more than 32 results in a small performance improvement (approximately less than 0.2%), but the training time increases linearly with the ensemble size.

C. RF Classifier

The RF model (as described above) was applied on the Hoda dataset. Both of the proposing preprocessing algorithms and feature representations with varying number of tree size were examied. For each different setup, the experiments were performed for *5 times* and the *mean* and *standard deviation* were reported. The obtained results are presented in Figure 2. For completeness, the details of performance and run time of all different setups of RF experiments are listed in Table III.

TABLE II. SUMMARY OF RESULTS FROM BEST PERFORMING SETUP OF EACH CATEGORY OF MODELS.

System	Accuracy*	Time+
Linear SVM, Padding, C = 1	89.91	265
1-NN, Padding, Uniform weights, Euclidean distance	96.19	1080
RF, Padding, HOG, Tree size = 256 (ours)	**98.12 (±0.03)**	**5538**
CNN, Padding, LeNet network, GPU training (ours)	**99.03 (±0.03)**	**80**

*Percent +Second

D. CNN Classifier

A CNN similar in architecture to [9] was also used for the task. Notice that CNN uses raw pixel values and learns multiple layer of feature extractors and a classifier in an end-to-end fasion from labeled data. The details of performance and run time of the CNN experiments are listed in Table II.

E. Summary of Results and Analysis

In summary, the best performing systems from each category in Table II were included. The performance of our system can further be analyzed by producing its confusion matrix. Figures 4 and 5 show the confusion matrix of the proposed RF (*256 trees*, *HOG features*, and *padding* preprocessing) and the CNN model. One can see that large confusion is presented among digits 0-5, 2-3, 3-4, and 6-9. This was obvious to us from the beginning as these digits have similarappearances in Persian handwriting. Some misclassified samples are shown in Figure 3.

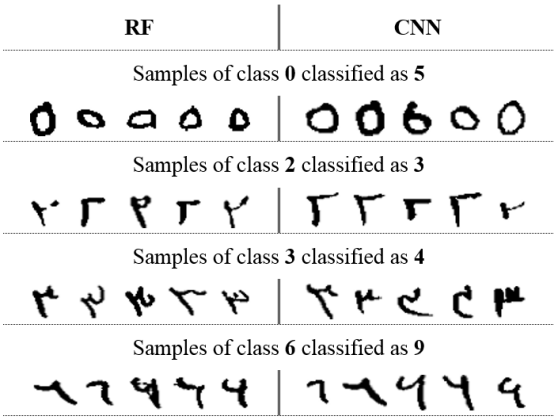

Fig. 3. Some misclassified samples from the test dataset of Hoda. As one can see in Persian handwriting, these letters sometimes are very similar.

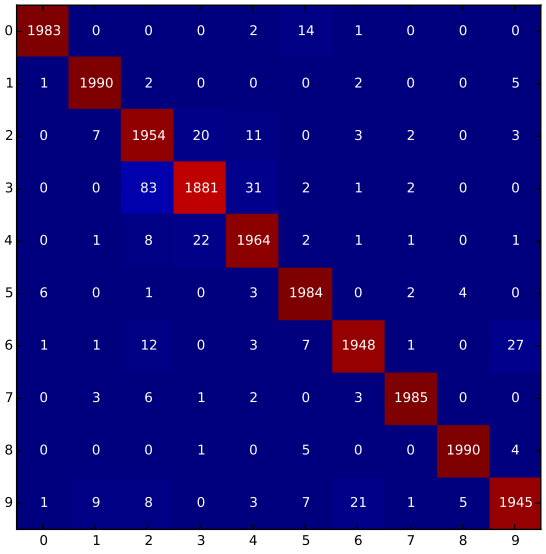

Fig. 4. Confusion matrix of RF model with 256 trees, HOG features, and padding preprocessing.

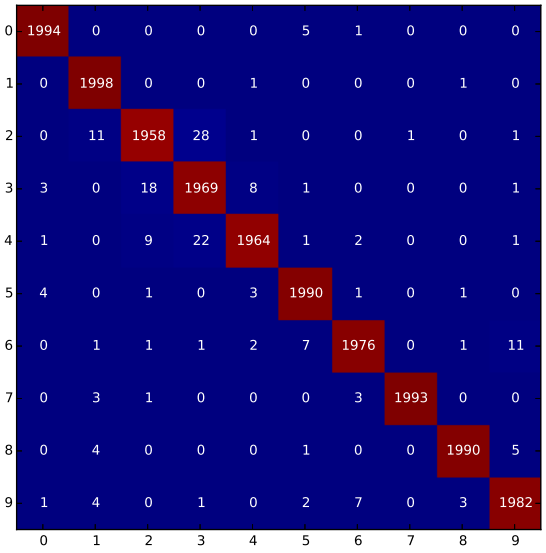

Fig. 5. Confusion matrix of CNN-LeNet with padding preprocessing.

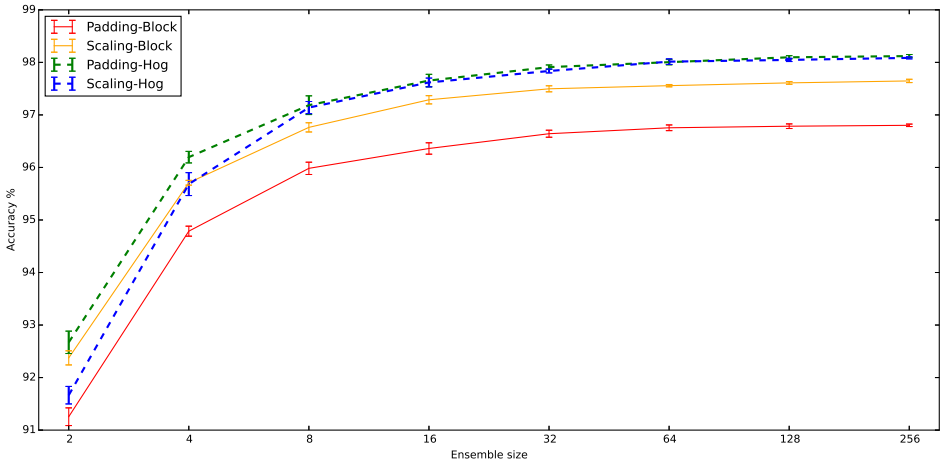

Fig. 2. Performance measure of RF classifier on Hoda dataset with varying ensemble size.

TABLE III. DETAILS OF PERFORMANCE AND RUN TIME OF DIFFERENT SETUPS OF PROPOSED RF METHOD. FOR EACH FEATURE TYPE AND SIZE, THE TIME CONSUMED DURING TRAINING PHASE IS REPORTED.

Ensemble Size	Scaling Block*	Padding Block*	Time$^+$	Scaling HOG*	Padding HOG*	Time$^+$
8	96.76 (0.09)	95.98 (0.12)	21	97.14 (0.12)	97.19 (0.18)	171
16	97.29 (0.08)	96.36 (0.11)	42	97.62 (0.08)	97.65 (0.12)	335
32	97.50 (0.06)	96.64 (0.07)	84	97.84 (0.04)	97.91 (0.04)	665
64	97.56 (0.02)	96.76 (0.05)	180	98.01 (0.05)	98.00 (0.05)	1344
128	97.61 (0.03)	96.79 (0.04)	345	98.05 (0.03)	98.10 (0.03)	2650
256	97.65 (0.03)	96.80 (0.03)	705	98.09 (0.02)	98.12 (0.03)	5538

*Percent $^+$Second

IV. CONCLUSION

An efficient Persian handwritten digit recognition method based on random forest and convolutional neural network was presented. Extensive experiments with various baselines were performed on the Hoda dataset. The proposed RF method performed comparable to the state-of-the-art methods. It was also fast when used with proper features; like HOG. Also, the CNN method achieved the state-of-the-art performance. In future, we will investigate other feature types and preprocessing stages that will close the gap on RFs and the state-of-the-art methods on this dataset.

REFERENCES

[1] A. Amin, "Off-line arabic character recognition: the state of the art," *Pattern recognition*, vol. 31, no. 5, pp. 517–530, 1998.

[2] H. Parvin, H. Alizadeh, M. Moshki, B. Minaei-Bidgoli, and N. Mozayani, "Divide & conquer classification and optimization by genetic algorithm," in *IEEE Third International Conference on Convergence and Hybrid Information Technology, 2008. ICCIT'08.*, vol. 2, 2008, pp. 858–863.

[3] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, "A new approach to improve the vote-based classifier selection," in *IEEE Fourth International Conference on Networked Computing and Advanced Information Management, 2008. NCM'08.*, vol. 2, 2008, pp. 91–95.

[4] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli, and M. Analoui, "A scalable method for improving the performance of classifiers in multiclass applications by pairwise classifiers and ga," in *IEEE Fourth International Conference on Networked Computing and Advanced Information Management, 2008. NCM'08.*, vol. 2, 2008, pp. 137–142.

[5] A. Alaei, U. Pal, and P. Nagabhushan, "Using modified contour features and svm based classifier for the recognition of persian/arabic handwritten numerals," in *IEEE Seventh International Conference on Advances in Pattern Recognition, 2009. ICAPR'09.*, 2009, pp. 391–394.

[6] M. Hamidi and A. Borji, "Invariance analysis of modified c2 features: case studyhandwritten digit recognition," *Machine Vision and Applications*, vol. 21, no. 6, pp. 969–979, 2010.

[7] A. Alaei, P. Nagabhushan, and U. Pal, "Fine classification of unconstrained handwritten persian/arabic numerals by removing confusion amongst similar classes," in *IEEE 10th International Conference on Document Analysis and Recognition, 2009. ICDAR'09.*, 2009, pp. 601–605.

[8] 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 (IJDAR)*, vol. 16, no. 4, pp. 371–386, 2013.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 161–168.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] 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, no. 10, pp. 1133–1141, 2007.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.