ÖHÜ Müh. Bilim. Derg. / OHU J. Eng. Sci.

ISSN: 2564-6605

doi: 10.28948/ngumuh.443160

Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, Cilt 7, Sayı 2, (2018), 504-513

Omer Halisdemir University Journal of Engineering Sciences, Volume 7, Issue 2, (2018), 504-513

Araştırma / Research

A COMPARATIVE PERFORMANCE ANALYSIS OF VARIOUS CLASSIFIERS FOR FINGERPRINT RECOGNITION

Alper BAŞTÜRK[1] (ORCID: 0000-0001-5810-0643)[*]
Nurcan SARIKAYA BAŞTÜRK[2] (ORCID: 0000-0002-5703-8355)
Orxan QURBANOV[3] (ORCID: 0000-0002-1298-8445)

*[*1]Erciyes Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kayseri, Türkiye*
[2]Erciyes Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, Uçak Elektrik Elektronik Bölümü; Kayseri, Türkiye
[3]Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD, Kayseri, Türkiye

Geliş / Received: *06.06.2018*
Kabul / Accepted: *20.06.2018*

ABSTRACT

In this study, recognition of fingerprint images has been performed by recent classifiers as well as some important and common classifiers available in the literature. The classification methods used in the study are support vector machines, k-nearest neighbors, Naive-Bayes, decision tree learning, and deep neural networks. Training/testing data set has been obtained basically by using four different versions of fingerprint images of 165 different fingers. Additional seven rotated versions of each different fingerprint images are also used to extend the data set. Feature vector of each fingerprint image (a fingercode) has been produced by using directional Gabor filters and averaging specific regions (sectors) of their output images. After creating fingercode data set, all classifiers has been trained to recognize fingerprint images. Detailed simulation results show that deep neural networks can be effectively used among all classifiers for recognition of fingerprint images.

Keywords: Deep neural networks, fingerprint recognition, classification

PARMAK İZİ TANIMA İÇİN FARKLI SINIFLANDIRICILARIN KARŞILAŞTIRMALI BAŞARIM ANALİZİ

ÖZ

Bu çalışmada, güncel sınıflandırıcılar ve ayrıca literatürdeki mevcut bazı önemli ve yaygın sınıflandırıcılar kullanılarak parmak izi görüntüleri tanınmıştır. Çalışmada kullanılan sınıflandırma yöntemleri; destek vektör makineleri, k-en yakın komşu, Naive-Bayes, karar ağacı öğrenimi ve derin sinir ağlarıdır. Eğitim ve test veri setleri temel olarak 165 farklı parmağın dört farklı parmak izi görüntüsü alınarak elde edilmiştir. Her bir farklı parmak izi görüntüsüne ek olarak, bu izlerin yedi farklı döndürülmüş versiyonu da veri kümesini genişletmek amacıyla kullanılmıştır. Her parmak izi görüntüsünün özellik vektörü (parmak kodu), yönlü Gabor süzgeci ile süzgeçleme sonrası çıktı görüntülerindeki ilgilenilen (sektör) alanlarının ortalaması alınarak üretilmiştir. Parmak izi veri seti oluşturulduktan sonra, tüm sınıflandırıcılar parmak izi görüntülerini tanımak üzere eğitilmiştir. Detaylı simülasyon çalışmaları, parmak izi görüntülerinin tanınması amacıyla kullanılan sınıflandırıcılar arasında en başarımlı olanının derin sinir ağı tabanlı sınıflandırıcı olduğunu göstermiştir.

Anahtar kelimeler: Derin sinir ağı, parmak izi tanıma, sınıflandırma

1. INTRODUCTION

Quantifiable and unique traits of biological characteristics of beings are called as biometric data. Biometrics is the scientific discipline that includes the acquisition, measurement and recognition of biometric data. Biometric

[*] Corresponding author / Sorumlu yazar. Tel.: +90 352 207 66 66 Ext.:32552; e-mail: ab@erciyes.edu.tr

systems can be defined as the automated systems that process biometric data and can identify and/or match identities. Such systems stand out as more reliable systems since they have ability to distinguish and identify any phenomenon from others using unique traces of that phenomenon.

Nowadays, there are quite a number of biometric systems that can identify people by evaluating many different introductory features of them. Commonly encountered ones among these systems use physical properties of peoples such as face, fingerprint, iris, retina and hand geometry or characteristic features such as signature, sound, keystroke and etc. Among these biometric features, fingerprint is obviously the foreground one because it is most studied and employed trait of human being. A fingerprint can be represented by a number of features such as ridge pattern, ridge frequencies, locations of singular points (center and delta points), positions of distinctive points, number of ridges between detail pairs, and locations of pores etc. All these distinctive features are factors that ensure the individuality and uniqueness of the fingerprint and are stable since they have negligible temporal variability by time.

In the literature, classification methods for fingerprint recognition are mainly grouped as model-, structure-, syntactic-, statistical-, frequency-, computational intelligence-based and hybrid methods. In model-based methods, the singular points of the fingerprint and the positions of these points are addressed. This approach is often confronted as the method followed by fingerprint recognition experts [1]. The disadvantages of this method are the difficulties in detecting singular points in low-quality traces and consistency loss in the absence of singular points [2], [3]. Structure-based methods are studies in which the global ridge topology is analyzed, including the positions of singular points and the distributions of the ridge orientations. Structure-based methods are considered to be more stable methods than model-based methods because they use ridge orientation data instead of singular points information [4]. These methods include homogeneous maps based [5], dynamic mask template based [6], and hidden Markov model-based methods [7]. Syntactic-based methods are based on grammar statements that represent different class rules that are defined relative to the directional components in fingerprints. Different grammars such as tree grammar [7], [8] and stochastic grammar [9] are available in the literature. A study by Chang and Fan uses a combination of structure- and syntactic-based methods [10]. Another approach proposed in the literature (computational intelligence-based) is that classification of fingerprints can be regarded as a learning problem [11]. Many network structures, especially multi-layer perceptron (MLP) networks, are presented in the literature. In this context, the underlying work of such systems is the multi-layered network architecture offered by NIST [12]. An improved version of this work was presented at [13]. Kamijo et al. trained different MLP structures for this purpose [14]. Jain and his colleagues proposed a k-nearest neighbors decision rule and a neural network based classifier [4]. Marcialis and his colleagues used two classifiers together to increase the recognition performance [15]. Nagaty, and also Halici and Ongun have used self-organizing map structures [11], [16]. Nagaty's another work on this subject is based on back propagation networks [17]. Unlike single method approaches, hybrid methods use multiple methods together. With the popularization of deep learning, this method has also begun to be used in the literature to recognize fingerprints [18]–[21].

Deep learning has been used successfully in space researches, military defense systems, robotics, biomedical, big data analysis, pharmaceutical industry, voice and image synthesis, virtual reality applications and many other fields and subjects [22]–[25]. In fact, although the progress of deep learning-based methods cannot be described as entirely new technologies, their usage has been an increasing and huge trend for only a decade today arising from the new and amazing achievements of computational capabilities of computational systems. There is now almost no field or problem that they are unused. Nearly 30 years after than Professor John McCarthy's introduction the term ``artificial intelligence'' to the scientific literature firstly in 1955, Professor Geoffrey E. Hinton's publication in 1986 in Nature is introduced how artificial neural networks could be trained with backpropagation algorithm [26]. With this development, a tremendous increase and progress has been made in scientific studies on artificial neural networks. But the capabilities of the computational systems were still not at the desired level in those days, and this constituted the greatest obstacle in the use of structures with a large number of and specialized layers and too many neurons. As a matter of fact, in 1998, Yann LeCun revealed a gradient-based convolutional neural network architecture capable of successfully recognizing the digits 0-9 [27]. With the 2000s, especially after the Rajat Raina's work on the year 2009 on deep learning and the usage of graphics processors [28], the increasing ability of graphics processors has been assured the exceeding all computational limitations and finally today's scientific levels are reached.

In this study, 660 different fingerprint images composed by including 4 different fingerprint samples from 165 different fingers are recognized with various methods such as support vector machines (SVMs), k-nearest neighbors (KNN), Naive-Bayes (NB), decision tree (DT) learning, and deep neural networks (DNNs). Through this paper, we extend our work in [21] to compare the performances of these classifiers with each other. The classification is performed through the fingercode signals obtained from the fingerprint images. Detailed simulation results show that fingerprints can be effectively recognized using deep neural networks and fingerprint codes. In Section 2, information on the competing methods used in the study is given. Section 3 describes the simulation results and discussion. Finally, Section 4 presents the conclusion and remarks.

2. COMPETING METHODS

In this paper, five different methods, SVM, KNN, NB, DT, and DNN, are used for fingerprint image recognition and performances of these methods are compared. Details of these methods are given in the following subsections below:

2.1. Support Vector Machines Classifier

SVMs, one of the machine learning approaches, are models that are trained in a supervised fashion using the input-labeled output data pairs [29]. After an appropriate training procedure, SVM models can predict the corresponding class of a data they have not previously encountered. They have been successfully used for binary classification problems. When multiple class classification required problems need to be solved, however, one-to-many or one-to-one approaches can be used [30]. In one-to-many approach, SVMs as many as the number of classes are used together. In one-to-one approach, SVMs are trained for each binary class tuples and classifications are made based on which class receives the most support. From this respect, one-to-many approach can be considered as more cost-effective than the other.

2.2. K-Nearest Neighbors Classifier

The KNN algorithm is one of the most commonly used simple and effective pattern classification methods. In this classification approach, class of the data and the closeness of neighborhood are determined by the value of k [31]. KNN determines neighborhood weights according to contributions to the average of neighboring points. For this reason, the average value of the contributions of nearby neighbors are greater than those of more distant ones.

2.3. Naive-Bayes Classifier

Based on the simplified Bayes independence proposition, NB classifier is a probabilistic approach that uses the supervised training method. An NB classifier can also be thought of as a Bayes network in which each feature is conditionally independent of each other and the phenomenon intended to be learned is conditionally bound to all these features [32].

2.4. Decision Tree Learning Classifier

DT learning classifier, which is a predictive modelling approach, is one of the machine learning techniques. These models, which are structured in the form of a tree structure, are made up of leaves and branches [33]. In DT models, leaf elements represent class labels, while branch elements represent conjunctions of features that lead to those class labels. DTs have many advantages, such as being more understandable and interpretable, less needing to prepare the data (for example there is no need to normalization), logically more understandable (white box), and applicable for large data sets etc.

2.5. Deep Neural Networks

2.5.1. Autoencoder

An autoencoder (AE) network consists of an input layer, an encoder part, a decoder part and an output layer as shown in Figure 1. It is used to obtain the *code* signal $\mathbf{w}_c$ representing the network input $\mathbf{i}$. The AE is trained to produce the same $\mathbf{i}$ (ideally the same but it actually produces $\hat{\mathbf{i}}$) the signal at the decoder output for the input signal $\mathbf{i}$; i.e. the input and target output signals are chosen as the same. Thus, owing to training which will allow the input symbol fed to the network to be obtained at the decoder output, the code data that contains the new attributes representing the original input data is produced at the output of the encoder section as $\mathbf{w}_c$ [34]–[36]. This procedure is shown in Figure 1:

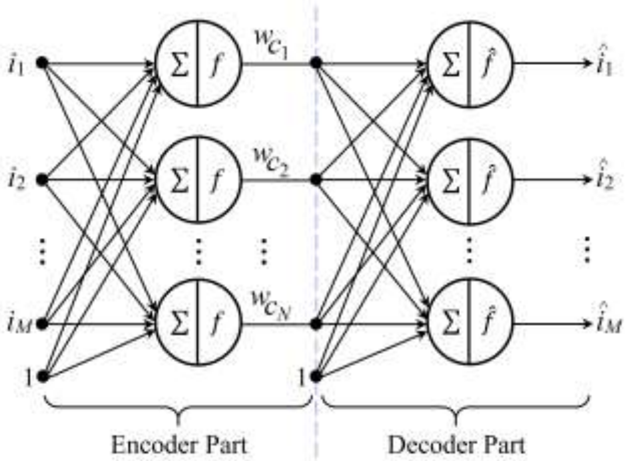

Figure 1. Typical structure of an autoencoder.

The encoder output of the AE can be expressed similarly as:

$$\mathbf{w}_c = f\left(\mathbf{b} + \mathbf{W}^{\mathbf{T}}\mathbf{i}\right) \tag{1}$$

Here, $\mathbf{w}_c = [w_{c_1}\ w_{c_2}\ \dots\ w_{c_N}]^T$ is the encoder output, $\mathbf{i} = [i_1\ i_2\ \dots\ i_M]^T$ is the input, $\mathbf{b} = [b_1\ b_2\ \dots\ b_N]^T$ is the bias weights of the neurons in the encoder part, $\mathbf{W} = [\mathbf{W}_1\ \mathbf{W}_2\ \dots\ \mathbf{W}_N]$ is the weights of the connections between the inputs and the neurons in the encoder part, f represents the activation function of neurons. For this structure, numbers of the inputs $\mathbf{i}$, outputs $\hat{\mathbf{i}}$ and the neuron cells in the encoder part are M, M and N, respectively.

The output of the AE can be similarly expressed as:

$$\hat{\mathbf{i}} = \hat{f}\left(\hat{\mathbf{b}} + \hat{\mathbf{W}}^T\mathbf{w}_c\right) \tag{2}$$

where $\hat{\mathbf{i}} = [\hat{i}_1\ \hat{i}_2\ \dots\ \hat{i}_M]^T$ is the output of the decoder part (and also of the AE), $\hat{\mathbf{b}} = [\hat{b}_1\ \hat{b}_2\ \dots\ \hat{b}_M]^T$ is the bias weights of the neurons in the decoder part, $\hat{\mathbf{W}} = [\hat{\mathbf{W}}_\mathbf{1}\ \hat{\mathbf{W}}_\mathbf{2}\ \dots\ \hat{\mathbf{W}}_\mathbf{M}]$ represents the weights of the connections between the output of the neurons on the encoder part and the neuron cells in the decoder part.

The cost function used in the training of the AE can be given by Eq. (3):

$$E = E_S + \beta\sum_{j=1}^{N}\nabla\left(\rho\middle\|\hat{\rho}_j\right) \tag{3}$$

Here, E_S is calculated by Eq. (4) and corresponds to the objective function for classical artificial neural networks:

$$E_S = \frac{1}{S}\sum_{k=1}^{S}e_k^2 + \frac{\lambda}{2}\left(\|\mathbf{W}\| + \|\hat{\mathbf{W}}\|\right) \tag{4}$$

where the λ term in Eq. (4) is a regularization term (also known as a weight decay term). The e_k term in the same expression, which represents the difference between the desired output and the calculated network output, can be calculated by Eq. (5):

$$e_k = \left\| \mathbf{i}^{(k)} - \hat{\mathbf{i}}^{(k)} \right\| \tag{5}$$

Here, for S sample values, $k = 1 \cdots S$ represents the sample counter.

The expression $\sum_{j=1}^{N} \nabla\left(\rho \| \hat{\rho}_j\right)$ in the cost function in Eq. (3) is called as the penalty term, and the β parameter in $\beta \sum_{j=1}^{N} \nabla\left(\rho \| \hat{\rho}_j\right)$ controls the weight of the sparsity penalty term. The relationship between the sparse parameter ρ and the mean activation value $\hat{\rho}_j$ is called as the Kullback-Leibler (KL) divergence (∇). KL divergence ∇ for the AE can be calculated by using Eq. (6) [35], [36]:

$$\nabla\left(\rho \| \hat{\rho}_j\right) = \rho log \frac{\rho}{\hat{\rho}_j} + \left(1-\rho\right) log \frac{1-\rho}{1-\hat{\rho}_j} \tag{6}$$

and

$$\hat{\rho}_j = \frac{1}{S} \sum_{k=1}^{S} f_j\left(\mathbf{i}^{(k)};\ \mathbf{W},\ \mathbf{b}\right). \tag{7}$$

2.5.2. Softmax Classifier

Softmax (SM) is a generalized version of the logistic regression which is a statistical method and widely used in binary classification. SM can classify data that have two or more classes. For this reason, they can be used as the final layer for AE structures to match two or more classes.

Training of SM structures is performed in a supervised manner. That is, for each sample in training input data set, what are the real classes of these samples should be known before the training. The trained SM estimates the appropriate classes for input samples with a process based on probabilistic computations [37]. SM structures are often encountered in deep models because they can be easily integrated into AEs [38].

3. SIMULATION RESULTS AND DISCUSSION

The proposed DNN model for fingerprint recognition has three cascade network layers. The first two of these layers are AEs, and the last layer is a SM. The scaled conjugate gradient (SCG) algorithm [39] is chosen for training the proposed DNN structure. There are 384 and 256 neurons in the first and second AE layers, respectively. The proposed system is shown in Figure 2.

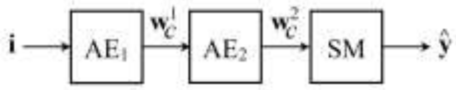

Figure 2. The proposed DNN structure.

The SCG algorithm has been run for 2000 epochs for the training of AE and SM layers and 5000 epochs for the fine-tuning phase of the whole DNN structure which is formed by integrating these three separate layers individually trained before. During the training, regularization weight ℓ_2, penalty term β and sparsity regularization parameter were chosen as 0.001, 4 and 0.05 by trial-and-error method, respectively.

The data needed to train the developed DNN structure was obtained from fingerprints of 165 different fingers. 4 different images of each finger were acquired. Thus, there are 660 fingerprint images in total. Each fingerprint was also subjected to additional rotation by seven times. Thus, a total of 5280 different fingerprint images have been reached. The purpose of this process is to ensure that the developed work is robust against the rotated form of the received fingerprint image and to increase the number of samples in the data set. Then, based on the approach in [4], each fingerprint image was filtered with 8 directional Gabor filters (0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°) and features for each image were obtained. When features were generated from the images, each output image of Gabor filters was divided into a total of 80 ROI fields with a circular grid pattern having 16 sectors and 5 bands in each sector. Then the average value of each ROI field was calculated and the *fingercode* includes these 80 average ROI field values was obtained for a single image. Thus, since each of the fingerprint images has been filtered by 8 Gabor filters and each Gabor filter outputs have 80 features (each fingerprint image has 640 features), the input data set used in the study has become a matrix of 640×5280. The developed DNN structure reduces these 640 attributes of each fingerprint to 384 by the first AE layer and 256 by the second AE layer. Thus, a mapping is made to 165 different classes via the SM layer for the features reduced from 640 to 256 by the first two AE layers. The developed DNN structure has been trained on the GPU and a 10-fold cross-validation approach has been adopted. The duration of a single training session lasts approximately 17 minutes on the Nvidia Geforce 980GTX GPU card. When the training is performed on the CPU, this 17-minute time is increased about 1.65 times and a 28-minute time period is required. The number of optimized internal parameters of the proposed DNN structure is (640×384)+(384×256)+(256×165)+384+256+165=387109.

The trained DNN structure and the competing methods were run 30 times to produce fair test results using different random test data sets. A total of 15959 fingerprint feature vectors have been randomly tested in 30 different runs. In this 15959 recognition test processes, 270, 555, 639, 6072 and 7544 false recognitions have been performed by the methods DNN, SVM, KNN, NB and DT, respectively. When the average values of these test performances are considered, the values for the methods are again encountered in the same order as 9.00, 18.30, 21.30, 202.40 and 251.47. When the obtained results are evaluated, it is understood that the competing methods performed the correct fingerprint recognition as 15689, 15404, 15320, 9887 and 8415 times, respectively. When the average recognition accuracy performances of the competing methods are examined, the values of 98.31, 96.53, 95.99, 61.94 and 52.73 are found, respectively. These results can be shown in Figs. 3 and 4 and also in Table 1. From Table 1, standard deviation results of the methods for misclassification and accuracy performances can also be seen. As can be seen from the Figs. 3.-4. and the Table 1., the most successful and consistent method among the competing methods is found as DNN.

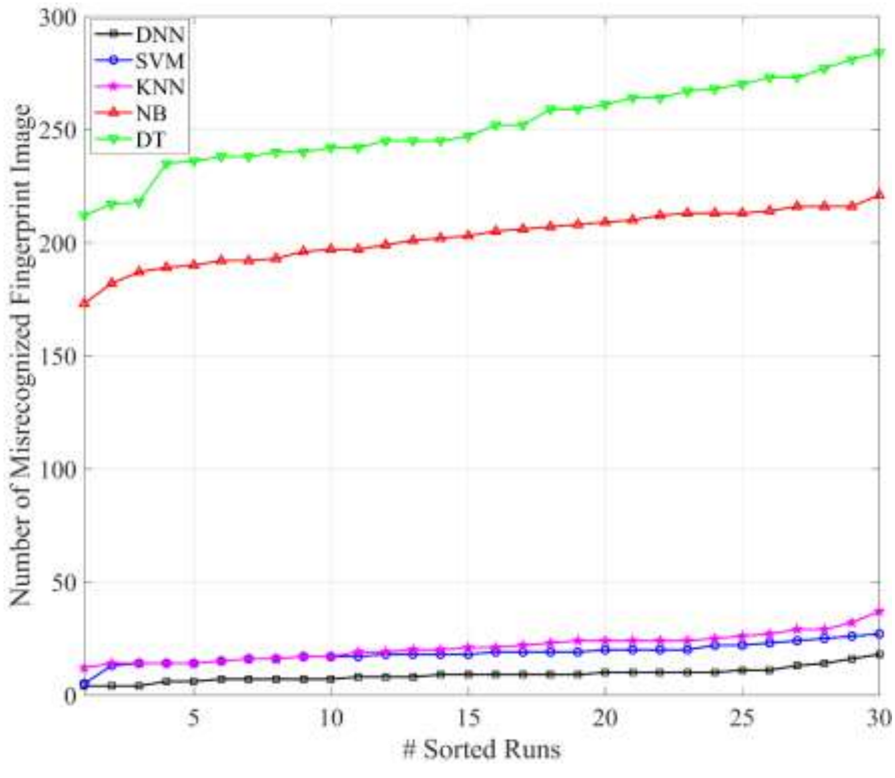

Figure 3. Sorted fingerprint recognition performances in terms of misrecognition count of the competing methods for 30 runs. (Less value indicates higher performance)

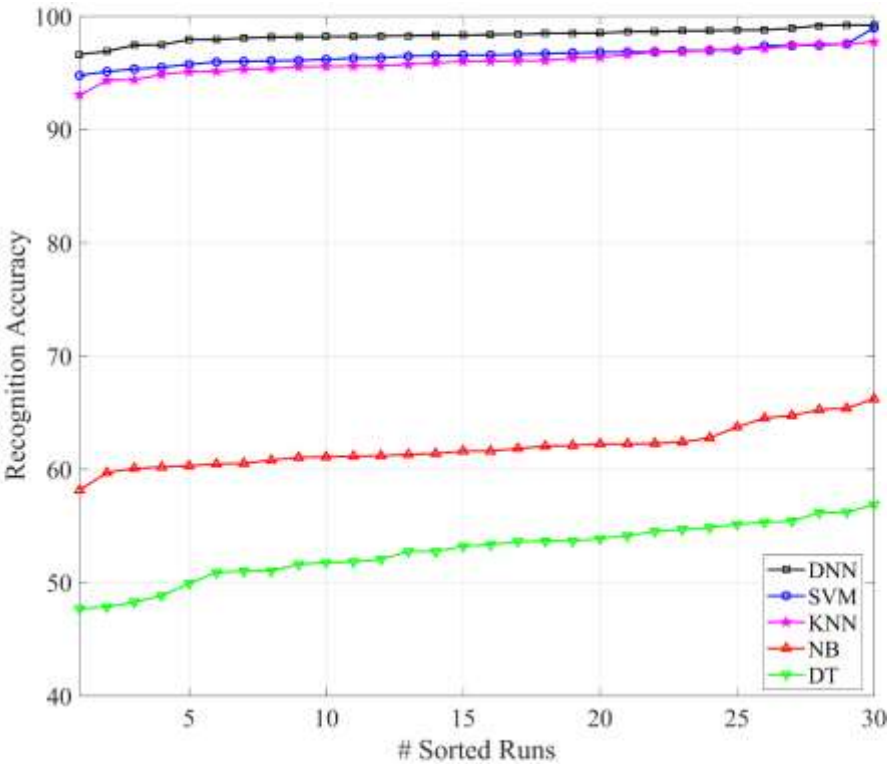

Figure 4. Sorted accuracy performances of the competing methods for 30 runs. (Higher value indicates higher performance)

Table 1. Average classification performances of the competing methods for 30 runs.

	Number of Misclassification		**Recognition Accuracy**	
Method	**Mean**	**Std**	**Mean**	**Std**
DNN [21]	9.00	3.22	98.31	0.60
SVM	18.30	4.40	96.53	0.83
KNN	21.30	5.97	95.99	1.08
NB	202.40	11.61	61.94	1.84
DT	251.47	18.77	52.73	2.52

In Figure 5, the performances of the methods are expressed through confusion matrices. These confusion matrices represent the obtained resultant fingerprint recognition performances of the methods for all of the 30 runs. For example, in Figure 5a, the diagonal of the confusion matrix represents 15689 correct recognition, while the non-diagonal outliers represent 270 fails. It can be easily seen from the Figure 5 that the best performance is of the DNN, again. Since all methods are trained using with 10-fold cross-validation technique for total 30 runs, the number 532 (531.97) corresponding to one out of thirty of the 15959 tests process and also corresponds to approximately one out of ten of the total of the 5280 fingerprint feature vectors. Thus, it is understood that DNN, which stands out as the most successful method, incorrectly classified 9 out of 532 samples on average. While this value can be regarded as a reasonable value, it will be useful to try to reduce this result by trying different DNN structures and / or approaches.

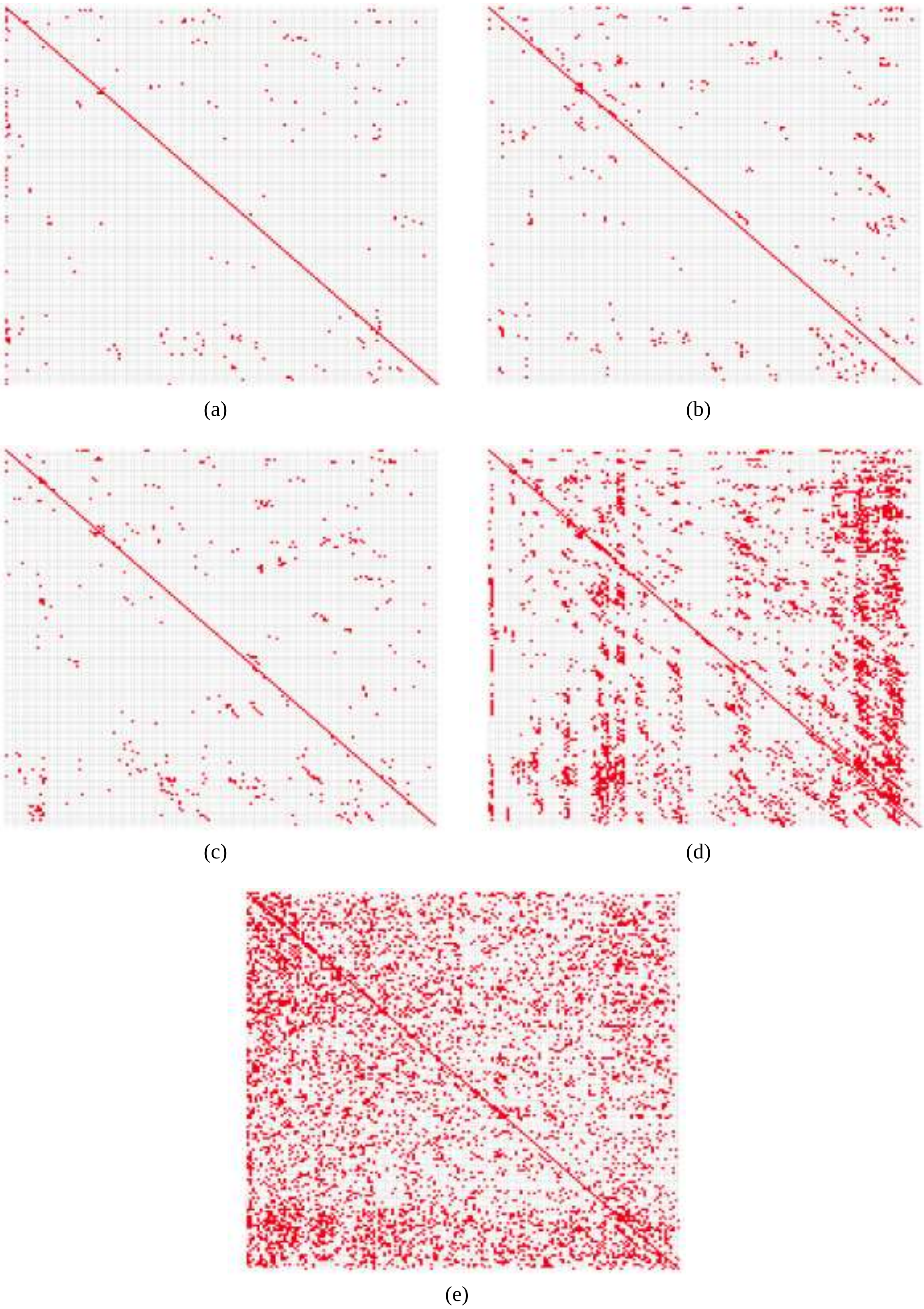

Figure 5. Resultant confusion matrices for 30 runs for all methods, a) DNN, b) SVM, c) KNN, d) NB, e) DT.

4. CONCLUSION

The fingerprint feature code, fingercode, is used effectively in the literature. In this study, it was also shown that if the fingerprint feature codes are selected as the training data for deep neural networks, fingerprints can be recognized with them, so that the related identities can be detected. The deep neural network structure used in the study is formed by two autoencoders and a softmax classifier. The scaled conjugate gradient algorithm is used for training the networks and the training process is performed on the graphics processor instead of the CPU for shortening the training period required. Detailed simulation results have shown that deep neural networks can recognize fingerprints with low error rates and can extract fewer new attributes from a large number of attributes. In the prospective studies, it is planned to achieve more successful results by using different deep neural networks and / or deep learning structures, different algorithms, different attributes and feature extraction methods.

ACKNOWLEDGEMENT

This work is supported by the Erciyes University Scientific Research Projects Coordination Unit (ERU/BAP) under project code FYL-2017-7229.

REFERENCES

[1] NALINI K. RATHA, R. BOLLE, Automatic fingerprint recognition systems. Springer, 2004.

[2] Q. ZHANG, H. YAN, “Fingerprint classification based on extraction and analysis of singularities and pseudo ridges,” Pattern Recognit., vol. 37, no. 11, pp. 2233–2243, Nov. 2004.

[3] J. LI, W.-Y. YAU, H. WANG, “Combining singular points and orientation image information for fingerprint classification,” Pattern Recognit., vol. 41, no. 1, pp. 353–366, Jan. 2008.

[4] A. K. JAIN, S. PRABHAKAR, LIN HONG, “A multichannel approach to fingerprint classification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 4, pp. 348–359, Apr. 1999.

[5] D. MAIO, D. MALTONI, “A structural approach to fingerprint classification,” in Proceedings of 13th International Conference on Pattern Recognition, 1996, pp. 578–585 vol.3.

[6] R. CAPPELLI, A. LUMINI, D. MAIO, D. MALTONI, “Fingerprint classification by directional image partitioning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 5, pp. 402–421, May 1999.

[7] A. SENIOR, “A combination fingerprint classifier,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pp. 1165–1174, 2001.

[8] B. MOAYER, K.-S. FU, “A Tree System Approach for Fingerprint Pattern Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 3, pp. 376–387, May 1986.

[9] B. MOAYER, K. S. FU, “An application of stochastic languages to fingerprint pattern recognition,” Pattern Recognit., vol. 8, no. 3, pp. 173–179, Jul. 1976.

[10] J.-H. CHANG, K.-C. FAN, “A new model for fingerprint classification by ridge distribution sequences,” Pattern Recognit., vol. 35, no. 6, pp. 1209–1223, Jun. 2002.

[11] K. A. NAGATY, “Fingerprints classification using artificial neural networks: a combined structural and statistical approach,” Neural Networks, vol. 14, no. 9, pp. 1293–1305, Nov. 2001.

[12] C. L. WILSON, G. T. CANDELA, C. I. WATSON, “Neural network fingerprint classification,” J. Artif. Neural Networks, vol. 1, no. 2, pp. 203–228, 1994.

[13] WILSON, C. L., BLUE, J. L., OMIDVAR O. M., “Improving Neural Network Performance for Character and Fingerprint Classification by Altering Network Dynamics,” in Proceedings of theWorld Congress on Neural Networks, 1995, pp. 151–158.

[14] M. KAMIJO, H. MIENO, K. KOJIMA, “Classification of fingerprint images using a neural network,” Syst. Comput. Japan, vol. 23, no. 3, pp. 89–101, 1992.

[15] G. LUCA MARCIALIS, F. ROLI, P. FRASCONI, “Fingerprint Classification by Combination of Flat and Structural Approaches,” in International Conference on Audio-and Video-Based Biometric Person Authentication, Springer, Berlin, Heidelberg, 2001, pp. 241–246.

[16] U. HALICI, G. ONGUN, “Fingerprint classification through self-organizing feature maps modified to treat uncertainties,” Proc. IEEE, vol. 84, no. 10, pp. 1497–1512, 1996.

[17] K. A. NAGATY, “On learning to estimate the block directional image of a fingerprint using a hierarchical neural network,” Neural Networks, vol. 16, no. 1, pp. 133–144, Jan. 2003.

[18] L. N. DARLOW, B. ROSMAN, “Fingerprint minutiae extraction using deep learning,” in 2017 IEEE

International Joint Conference on Biometrics (IJCB), 2017, pp. 22–30.

[19] D. MICHELSANTI, A.-D. ENE, Y. GUICHI, R. STEF, K. NASROLLAHI, T. B. MOESLUND, "Fast Fingerprint Classification with Deep Neural Networks," in Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, 2017, pp. 202–209.

[20] Y. WANG, Z. WU, J. ZHANG, "Damaged fingerprint classification by Deep Learning with fuzzy feature points," in 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2016, pp. 280–285.

[21] A. BASTURK, N. SARIKAYA BASTURK, O. QURBANOV, "Fingerprint Recognition by Deep Neural Networks and Fingercodes," in IEEE 26th Signal Processing and Communications Applications Conference, 2018, pp. 1–4.

[22] H. BADEM, A. BASTURK, A. CALISKAN, M. E. YUKSEL, "A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited–memory BFGS optimization algorithms," Neurocomputing, vol. 266, pp. 506–526, Nov. 2017.

[23] A. CALISKAN, H. BADEM, A. BAŞTÜRK, M. E. YÜKSEL, "Diagnosis of the parkinson disease by using deep neural network classifier," Istanbul Univ. - J. Electr. Electron. Eng., vol. 17, no. 2, pp. 3311–3318, 2017.

[24] A. CALISKAN, M. E. YUKSEL, H. BADEM, A. BASTURK, "A Deep Neural Network Classifier for Decoding Human Brain Activity Based on Magnetoencephalography," Elektron. ir Elektrotechnika, vol. 23, no. 2, pp. 63–67, Apr. 2017.

[25] A. CALISKAN, M. E. YUKSEL, H. BADEM, A. BASTURK, "Performance improvement of deep neural network classifiers by a simple training strategy," Eng. Appl. Artif. Intell., vol. 67, pp. 14–23, Jan. 2018.

[26] D. E. RUMELHART, G. E. HINTON, R. J. WILLIAMS, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[27] Y. LECUN, L. BOTTOU, Y. BENGIO, P. HAFFNER, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] R. RAINA, A. MADHAVAN, A. Y. NG, "Large-scale deep unsupervised learning using graphics processors," in Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, 2009, pp. 873–880.

[29] T.-F. WU, C.-J. LIN, R. C. WENG, "Probability Estimates for Multi-class Classification by Pairwise Coupling," J. Mach. Learn. Res., vol. 5, pp. 975–1005, 2004.

[30] M. ARUN KUMAR, M. GOPAL, "Reduced one-against-all method for multiclass SVM classification," Expert Syst. Appl., vol. 38, no. 11, pp. 14238–14248, Oct. 2011.

[31] M. E. HELLMAN, "The Nearest Neighbor Classification Rule with a Reject Option," IEEE Trans. Syst. Sci. Cybern., vol. 6, no. 3, pp. 179–185, Jul. 1970.

[32] R. R. YAGER, "An extension of the naive Bayesian classifier," Inf. Sci. (Ny)., vol. 176, no. 5, pp. 577–588, Mar. 2006.

[33] L. ROKACH, O. MAIMON, Data Mining with Decision Trees, vol. 69. WORLD SCIENTIFIC, 2007.

[34] M. RANZATO, C. POULTNEY, S. CHOPRA, Y. LECUN, "Efficient learning of sparse representations with an energy-based model," Proceedings of the 19th International Conference on Neural Information Processing Systems. MIT Press, pp. 1137–1144, 2006.

[35] Q. V LE, J. NGIAM, A. COATES, A. LAHIRI, B. PROCHNOW, A. Y. NG, "On Optimization Methods for Deep Learning," in Proceedings of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 265–272.

[36] Y. BENGIO, "Practical Recommendations for Gradient-Based Training of Deep Architectures," Springer, Berlin, Heidelberg, 2012, pp. 437–478.

[37] S. TAO, T. ZHANG, J. YANG, X. WANG, W. LU, "Bearing fault diagnosis method based on stacked autoencoder and softmax regression," in 2015 34th Chinese Control Conference (CCC), 2015, pp. 6331–6335.

[38] Y. ZHANG, E. ZHANG, W. CHEN, "Deep neural network for halftone image classification based on sparse auto-encoder," Eng. Appl. Artif. Intell., vol. 50, pp. 245–255, Apr. 2016.

[39] M. F. MØLLER, "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, no. 4, pp. 525–533, Jan. 1993.