

# Persian Handwritten Digit Recognition with Classifier Fusion: Class Conscious versus Class Indifferent Approaches

Reza Ebrahimpour, and Fatemeh Sharifzadeh

**Abstract**—A large experiment on Persian handwritten digits are reported and discussed. In this paper the techniques to combine multiple classifiers based on static structures is investigated. A static structure includes two main strategies to combine result of base classifiers: a) class indifferent methods b) class conscious methods. We establish our model on Decision Template and Dempster Shafer, which are under category of class indifferent method, and compare their recognition rate with five of the most famous combining methods of class conscious category. To evaluate our proposed model a real-world database of Persian handwritten digits containing 8600 handwritten digit images is used. Experiments using our database demonstrate that combining result of base classifiers with class indifferent methods indeed are far more effective than combining the result with class conscious methods in Persian handwritten digit recognition. Evaluating the proposed system with 2150 test samples the recognition rate of 91.98% is achieved.

**Keywords**—Class conscious, Class indifferent, Classifier fusion, Decision template, Dempster Shafer, Persian Handwritten Digit Recognition.

## I. INTRODUCTION

IN the last few decades, numerous methods have been proposed for machine recognition of handwritten characters, especially for languages such as English, Japanese and Chinese due to the popularity of language use. Particularly, handwritten numeral recognition has attracted much attention, and various techniques (pre-processing, feature extraction, and classification) have been proposed [32-36].

In contrast, for the recognition of Persian (Arabic) handwritten digits very few works are reported ([2]; [10]; [37]; [19]). And now research on Farsi(Persian) scripts and numerals is receiving increasing attention because a lot of data such as addresses written on envelopes; amount written on checks; names, addresses, identity numbers, and Rial values written on invoices and forms were written by hand and they had to be entered into the computer for processing.

Combining classifiers to achieve higher accuracy is an important research topic [1,5,31,34,29]. Essentially, the idea behind combining classifiers is based on the so-called divide-

and-conquer principle, according to which a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks [3].

There are two main strategies in combining classifiers: fusion (static structures) and selection (dynamic structures) [4]. In classifier fusion, it is supposed that each ensemble member is trained on the whole feature space [5, 6], whereas in classifier selection, each member is assigned to learn a part of the feature space [7,8,9]. This way, in the former strategy, the final decision is made considering the decisions of all members, while in the latter strategy, the final decision is made by aggregating the decisions of one or a few of experts [3,17].

In this paper, combining classifiers based on the fusion of outputs of a set of different classifiers have been proposed as a method of improving the recognition performance, increasing efficiency of classifying and raising reliability in the system. The method developed here is based on a set of  $c$  matrices called *decision templates* (DTs). DTs are a robust classifier fusion scheme that combines classifier outputs by comparing them to a characteristic template for each class. DT fusion uses *all* classifier outputs to calculate the final support for each class, which is in sharp contrast to most other fusion methods which use *only the support for that particular class* to make their decision.

The rest of this paper is organized as follows: In the coming section, describes briefly the Principle Component Analysis for feature extraction method. In Section III describes the proposed model in details. It is followed by the experimental results in Section IV. Finally, Section V draws conclusion and summarizes the paper.

## II. FEATURE EXTRACTION

In the first stage of our proposed model the Principle Component Analysis (PCA) was used, to avoid a high dimensional and redundant input space and optimally design and train the experts.

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression [11], and is a common technique for finding patterns in data of high dimension.

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data

Reza Ebrahimpour is with Department of Electrical Engineering, Shahid Rajaei University, Tehran, Iran (e-mail: rebrahimpour@srutu.edu).

F. Sharifzadeh is with Department of Computer Sciences, Tehran University, Tehran, Iran.

of high dimension, where the luxury of graphical representation is not available, PCA is the one of powerful tool for analyzing data.

The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information [12].

Suppose that  $T_1, T_2, \dots, T_M$  are the projection vector of training data set, and each of these vectors has  $N$  elements. The mean vector,  $A$ , is computed with the following equation:

$$A = \frac{1}{M} \sum_{m=1}^M T_m \tag{1}$$

Subtract the mean from each of the data dimensions.

$$X_m = T_m - A \quad 1 < m < M \tag{2}$$

The mean subtracted is the average across each dimension. If describe  $Y$  via equation (3) then covariance matrix  $C$  calculated with equation (4):

$$Y = [X_1 \ X_2 \ \dots \ X_M] \tag{3}$$

$$C = \frac{1}{M} \sum_{m=1}^M X_m X_m^T = \frac{1}{M} Y Y^T \tag{4}$$

Since the data is  $N$  dimensional, the covariance matrix will be  $N \times N$ . finally the eigenvectors and eigenvalues of the covariance matrix is calculated, and then the  $k$  most significant component are picked out and formed feature vector. Thus the PCA projection matrix projects input patterns from an  $N$ -dimensional image space to a  $K$ -dimensional subspace.

### III. CLASSIFIER FUSION

Combining classifiers is an approach to improve the performance in classification particularly for complex problems such as those involving limited number of patterns, high-dimensional feature sets, and highly overlapped classes [13].

Suppose  $D$  is a single classifier. Let  $x \in R^n$  be a feature vector and  $\{1, 2, \dots, c\}$  be the label set of  $c$  classes. It is assumed that all  $c$  degrees are in the interval  $[0, 1]$ , in other words,  $D : R^n \rightarrow [0, 1]^c$ . The output of  $D$  is signifying it by  $\mu_D(x) = [\mu_D^1(x), \dots, \mu_D^c(x)]$ .

The classifier outputs can be categorized into three levels: abstract level (unique class), rank level (rank order of classes), and measurement level (confidence scores of classes) [14].

So the decision of  $D$  that is assigned to  $x$  is typically done by the *maximum membership rule*:

$$D(x) = K \Leftrightarrow \mu_D^K(x) = \max\{\mu_D^i(x)\} \quad i = 1, \dots, c \tag{5}$$

Now let  $\{D_1, \dots, D_L\}$  be a set of classifier and  $\Omega = \{\omega_1, \dots, \omega_c\}$  be the set of class labels. Denote the output of the  $i$ th classifier as  $D_i(x) = [d_{i,1}(x), \dots, d_{i,c}(x)]^T$ , where  $d_{ij}(x)$  the support that classifier  $D_i$  gives to the supposition that  $x$  comes from class  $\omega_j$ . At the abstract level,  $D_i(x)$  has only one nonzero element corresponding to the decided class. The rank order can be converted to class scores such that  $d_{ij}$  is the number of classes ranked below  $\omega_j$ . At the measurement level,  $d_{ij}$  is the discriminant value (similarity or distance) or probability-like confidence of  $\omega_j$ . The measurement-level outputs can be easily reduced to rank level and abstract level.

Construct  $D_{ens}$ , the combination output of the  $L$  classifier as:

$$D_{ens}(x) = F(D_1(x), \dots, D_L(x)) = [\mu_D^1, \dots, \mu_D^c]^T \tag{6}$$

where  $F$  is called aggregation rule.

The  $L$  classifier outputs for an input pattern  $x$  can be arranged in a decision profile matrix ( $DP(x)$ ) as shown in the Fig. 1 [37]:

$$DP(x) = \begin{bmatrix} d_{11}(x) & \dots & d_{1j}(x) & \dots & d_{1c}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{i1}(x) & \dots & d_{ij}(x) & \dots & d_{ic}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{L1}(x) & \dots & d_{Lj}(x) & \dots & d_{Lc}(x) \end{bmatrix}$$

Fig. 1 Decision profile matrix for an input pattern  $x$ . each row in this matrix is the output of classifier  $D_i(x)$  and each column exhibits the supports from classifier  $D_1, D_2, \dots, D_L$

There are two general approaches to use  $DP(x)$  to find the overall support for each class and subsequently label the input  $x$  in the class with the largest support.

- Some methods calculate the support for class  $i$  ( $\mu_D^i(x)$ ) using only the  $i$ th column of  $DP(x)$ . Such methods that use the  $DP$  class-by-class will be called *class-conscious methods*.

Examples of class-conscious fusion operators are: average, sum, minimum, maximum, product, fuzzy integral, etc.

The choice of an aggregation method  $F$  depends on the explanation of  $d_{ij}(x)$ ,  $i=1, \dots, L$ ,  $j=1, \dots, c$  and also is related to characteristic of data.

- Another fusion method is to use *all of*  $DP(x)$  to calculate the support for each class. Fusion methods in this category will be called *class-indifferent*. Here we can use any classifier with decision profile matrices, as inputs and the class label  $D_{ens}(x)$  as the output. There are however some class-indifferent fusion strategies such as decision templates or Dempster Shafer methods that details of these methods used in our model are described in Section III.A and III.B, respectively.

Notice the difference between the class-conscious and class-indifferent groups of methods. The former use the context of the DP but disregard part of the information, using *only one column per class*, but in the latter methods use the whole DP but neglect the context.

In this paper, our model is established upon Decision Template and Dempster Shafer, which is under the category of class indifferent methods, and compare its performance with five of the most famous combining methods of class conscious (namely minimum, maximum, sum, product and average). These methods are briefly described in the following.

**Minimum Rule:** In this method, the output node that is the maximum value among the minimums of experts' outputs, determines the final decision.

**Maximum Rule:** In this method, the output node that is the maximum value among the maximums of experts' outputs, determines the final decision.

**Average method:** In this method, the final decision is made by averaging the experts' outputs.

**Product method:** In this method, the output node that is the maximum value among the multiplication of experts' outputs, determines the final decision.

*A. Proposed Model: Decision Template*

The idea of the decision templates (DT) combiner is to remember the most typical Decision Profile for each class  $\omega_j$ , called the decision template,  $DT_j$ , and then compare it with the current decision profile  $DP(x)$  using some similarity measure  $S$ . The closest match will label  $x$ .

Let  $X = \{x_1, \dots, x_N\}, x_i \in R^n$ , be the training data set that are belongs to the class set  $\Omega = \{\omega_1, \dots, \omega_c\}$ , and the  $D = \{D_1, \dots, D_L\}$  be a set of classifier.

So the Decision Profile matrix for each of particular  $x_i$  is the  $L \times c$  matrix.

Definition: The decision template  $DT_i$  for class  $i$  is the average of the decision profiles of the elements of the training set  $X$  labeled in class  $i$ . thus  $DT_i(X)$  of class  $i$  is the  $L \times c$  matrix  $DT_i(X) = [dt_i(k,s)(X)]$  whose  $(k, s)$ th element is computed by [15,16]:

$$dt_i(k,s)(X) = \frac{\sum_{j=1}^N Ind(x_j, i) d_{k,s}(x_j)}{\sum_{j=1}^N Ind(x_j, i)}, k = 1, \dots, L, s = 1..c \quad (7)$$

where  $Ind(x_j, i)$  is an indicator function with value 1 if pattern  $x_j$  is belonged to class  $\omega_i$ , and 0, otherwise[16]. To simplify the notation  $DT_i(Z)$  will be denoted by  $DT_i$ .

After constructing DT, in testing phase, When  $x \in R^n$  is submitted for classification, the DT scheme matches  $DP(x)$  to  $DT_i, i=1,2, \dots, c$ , and produces the soft class labels

$$\mu_{D_{ens}}^i(x) = S(DT_i, DP(x)), i=1, \dots, c, \quad (8)$$

where  $S$  is interpreted as a *similarity* measure. The higher the similarity between the decision profile of the current  $x$  ( $DP(x)$ ) and the decision template for class  $i$  ( $DT_i$ ), the higher the support for that class ( $\mu_{D_{ens}}^i(x)$ ). Notice the word 'similarity' is used in a broad sense, meaning 'degree of match' or 'likeness', etc.

Two measures of similarity are based upon [17]:

- The squared Euclidean distance (DT (E)). The ensemble support for  $\omega_j$  is

$$\mu_j(x) = 1 - \frac{1}{L \times c} \sum_{i=1}^L \sum_{j=1}^c [DT_j(i, k) - d_{i,k}(x)]^2 \quad (9)$$

where  $DT_j(i, k)$  is the  $(i, k)$ th entry in decision template  $DT_j$ . The outputs  $\mu_j$  are scaled to span the interval  $[0,1]$ , but this scaling is not necessary for classification purposes. The scaling coefficient  $1/(L \times c)$  and the constant 1 can be dropped. The class with the maximal support would be the same. If  $DP(x)$  and  $DT_j$  regards as vectors in the  $L \times c$ -dimensional intermediate feature space, the degree of support is the negative squared Euclidean distance between the two vectors. This calculation is equivalent to applying the nearest mean classifier in the intermediate feature space. While only the Euclidean distance in Eq. (9) was used, there is no reason to stop at this choice. Any distance could be used, for example, the Minkowski, Mahalanobis, and so on.

- A symmetric difference (DT(S)). Symmetric difference comes from fuzzy set theory [26,27]. The support for  $\omega_j$  is

$$\mu_j(x) = 1 - \frac{1}{L \times c} \sum_{i=1}^L \sum_{j=1}^c \max\{\min\{DT_j(i, k), (1 - d_{i,k}(x))\}, \min\{(1 - DT_j(i, k)), d_{i,k}(x)\}\} \quad (10)$$

Example: Illustration of the Decision Templates (DT) Combiner. Let  $c = 2, L = 3$ , and the decision templates for  $\omega_1$  and  $\omega_2$  be respectively

$$DT_1 = \begin{bmatrix} 0.85 & 0.15 \\ 0.91 & 0.09 \\ 0.88 & 0.12 \end{bmatrix} \quad DT_2 = \begin{bmatrix} 0.15 & 0.85 \\ 0.18 & 0.82 \\ 0.14 & 0.86 \end{bmatrix}$$

Assume that for an input  $x$ , the following decision profile has been obtained:

$$DP(x) = \begin{bmatrix} 0.23 & 0.77 \\ 0.86 & 0.14 \\ 0.21 & 0.79 \end{bmatrix}$$

The similarities and the class labels using DT(E) are:

$$\mu_D(x) = [\mu_1(x), \mu_2(x)] = [0.7214, 0.8421]$$

So pattern  $x$  is belonged to class  $\omega_2$ .

Decision templates are a class-indifferent approach because they treat the classifier outputs as a context-free set of

features. All class-conscious combiners are idempotent by design, that is, if the ensemble consists of L copies of a classifier D, the ensemble decision will be no different from the decision of D.

Several studies have looked into possible applications of DTs [22-25].

*B. Proposed Model: Dempster Shafer*

This technique is the one closest to the DT. Two combination methods, which take their inspiration from the evidence combination of Dempster–Shafer (DS) theory, are proposed in Refs. [18,28]. The method proposed in Ref. [18] is commonly known as the Dempster–Shafer combiner .

The classifier outputs  $\{D_i(x)\}$  are possibilistic. Instead of calculating the similarity between the decision template  $DT_i$  and the decision profile  $DP(x)$ , the DS algorithm goes further. The following steps are performed:

1. Let  $DT_j^i$  denote the ith row of decision template  $DT_j$ .

Denote by  $D_i(x)$  the (soft label) output of  $D_i$ , that is,  $D_i(x) = [d_{i,1}(x), \dots, d_{i,c}(x)]^T$  : the ith row of the decision profile  $DP(x)$ . Calculate the “proximity”  $\varphi$  between  $DT_j^i$  and the output of classifier  $D_i$  for the input x [18]

$$\varphi_{j,i}(x) = \frac{(1 + \|DT_j^i - D_i(x)\|^2)^{-1}}{\sum_k (1 + \|DT_k^i - D_i(x)\|^2)^{-1}} \quad (11)$$

where  $\|\cdot\|$  is any matrix norm. For example, the Euclidean distance can be used between the two vectors. Thus for each decision template L proximities is achieved.

2. Using Eq. (11), Calculate for every class,  $j = 1, \dots, c$ ; and for every classifier,  $i = 1, \dots, L$ , the following belief degrees:

$$b_j(D_i(x)) = \frac{\varphi_{j,i}(x) \prod_{k \neq j} (1 - \varphi_{k,i}(x))}{1 - \varphi_{j,i}(x) \prod_{k \neq j} (1 - \varphi_{k,i}(x))} \quad (12)$$

3. The final DS label vector with membership degrees has the components

$$\mu_j(x) = K \prod_{i=1}^L b_j(D_i(x)), j = 1, \dots, c, \quad (13)$$

where K is a normalizing constant.[17]

Example: Illustration of the Dempster–Shafer Method. The two decision templates and the decision profile are:

$$DP(x) = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix} \quad DT_1 = \begin{bmatrix} 0.6 & 0.4 \\ 0.8 & 0.2 \\ 0.5 & 0.5 \end{bmatrix} \quad DT_2 = \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix}$$

Using above equation the proximity matrix and the belief degrees are:

$$\varphi(x) = \begin{bmatrix} 0.4587 & 0.5000 & 0.5690 \\ 0.5413 & 0.5000 & 0.4310 \end{bmatrix}$$

$$B(x) = \begin{bmatrix} 0.2799 & 0.3333 & 0.4289 \\ 0.3898 & 0.3333 & 0.2462 \end{bmatrix}$$

Finally  $\varphi(x)$  ,  $B(x)$  membership degree matrix of pattern x with  $K=13.89$  are as follows:

$$\mu_D(x) = [\mu_D^1(x), \mu_D^2(x)] = [0.5558 \quad 0.4442]$$

Thus the DS combiner gives a slight preference to class  $\omega_1$  .

IV. EXPERIMENTAL RESULTS

To evaluate the performance of proposed model and also exhibit the advantage of using it in recognition of Farsi digits, it is compared with other fusion methods such as Sum ,Min, Max, Average and product aggregation rules on gathered dataset.

*A. Database*

For training and testing our system 8600 digit images written by 860 different persons was collected, where each person wrote each of 10 digits. These participants were selected among the undergraduate students from universities in Iran. The samples were divided into the train and test sets by considering the samples belonging to 645 persons as the train set and the samples belonging to other 215 persons as the test set. Among the collected samples, some were written incorrectly or they were written very unusually that one do not expect in the ordinary Persian handwriting.

All of samples are scanned at 300 dpi resolution and in the grayscale format. Train and test images are both resized to  $40 \times 40$  pixel images at first.

Some samples of 10 classes for training set and testing set are shown in Fig. 2(a), 2(b), respectively.



Fig. 2 (a) Samples of Farsi numerals from training set



Fig. 2 (b) Samples of Farsi numerals from testing set

In both Fig. 2(a) and Fig. 2(b), first row exhibits the images that are easily recognized by humans without any ambiguity. Images in this category are clear and unambiguous. They have all the necessary structural primitives, and have typical connectivity of the primitives. The second row presents images that humans have difficulty in identifying them because of noise, filled loop, cursive writing, over-segmentation or similarity of their primitives and structures, etc.

### B. Classifier Structures

First, in order to decrease computational load and to achieve high accuracy, dimensionality reduction was performed using principal component analysis (PCA). Thus in the first stage to take a decision about the number of PCA components, a Multi Layer Perceptron with 35 hidden neurons and 10 output nodes was used as a classifier. Table I displays the different experiments to determine about the number of PCA components on the dataset that didn't use in the test phase.

TABLE I  
RECOGNITION RATES OF DIFFERENT NUMBER OF PCA COMPONENT FOR THE  
BASE CLASSIFIERS

Number of input neurons	15	20	30	40	50
Recognition rate(%)	84.22	84.90	<b><u>85.63</u></b>	82.61	81.48

The highest recognition rate is typed in bold and also is underlined. Each result is the average of ten times.

It should be mentioned that during different experiments a 30-dimensional subspace turned out to be the optimal case. The PCA projection matrix projects digit patterns from a 1600-dimensional image space to a 30-dimensional subspace.

For this experiment a set of 4 classifiers are used. They were optimized for the particular application. In this way they illustrate well the differences between these classifiers, and, moreover, it serves better the aim to study the effects of combining classifiers of various performances. As argued in the [21], it is important to make the outputs of the classifiers comparable. Now the set of basic classifiers are going to be discussed.

The MLP is used as the base classifiers with one hidden layer, with the connecting weights estimated by the error back-propagation (BP) algorithm minimizing the squared error criterion. The MLPs of all experts had 30 input nodes for PCA components and 10 output nodes corresponding to ten digits. For diversifying base classifiers, the weights of MLP neural networks are initially set to small random values. In addition, different topologies for base classifiers are assumed. The MLP has learning parameters, such as number of epochs, estimated by fourfold cross validation on the training set. For each of single MLP, the training and testing phase for different topologies is repeated, such as 30:35:10, 30:40:10, 30:45:10, and 30:50:10, for 10 times.

Fig. 3 illustrates the performance of each expert for every of 10 class of the proposed model, on the unseen digit images

of the training, averaged over 10 runs. The bars denote the average recognition rates of experts, broken down by 10 classes. Note that in Fig. 3 the left most bar in each classifier corresponds to digit 0, class 1, and the right most bar point out to class 10, digit 9.

The results of our proposed method using different fusion methods are presented at Tables II. In this table classification accuracy is shown for the data set. Here only the % correct on the test sets is displayed, which have not been seen during training of either the individual classifiers or the second level fusion models.

The left half section of the table deals with the class indifferent methods applied on all 4 base classifiers. In these methods 10 decision template matrixes are calculated, which are corresponded to 10 ten class. In decision template method the Euclidean distance is used to make decision about similarity between each of test samples and its corresponding class. In Dempster- Shafer after decision template matrix computed, the proximity matrix, belief degrees and the membership degree matrix of patterns calculated. The best results of class indifferent methods are underlined.

It appears that the DS fusion method frequently scores a best result. The calculations that it involves however, are more complex than any of the DT schemes. In addition all combined results are better than the best individual classifier performance. For instance, when learning rate of the base classifier is 0.001, the best recognition rate of individual classifier is about 88 %, combining all classifiers using these fusion method, however, improves this result. This combination rule is thereby useful.

In the entire right half of the table the results of class conscious methods are shown. The best results over the 5 combining rules are underlined. Again all combining results are better than the results for individual classifiers.

The best results for each row are printed in bold. The first thing to notice from Table II is that combining the results of base classifiers with class indifferent methods is far more effective than combining the results with class conscious methods. Clearly using all classifier outputs to calculate the final support for each class is more useful than other fusion methods which use only the support for that particular class to make their decision. For combining the results of classifiers on the category of class conscious the product combination rule gives good results. Kittler [30] showed that a product combination rule especially improves the estimate of the posterior probability when posterior probabilities with independent errors are combined.

The accuracy of the combinations in our experimental result in the either second row or third row of Table II are not very high compared to recognition rate on the same data sets reported in first row of same Table. This is performed, because it does not confer special attention on designing the individual first level classifiers. In this study we were interested in comparing the second-level fusion schemes, and hence, the type of first-level classifiers was irrelevant. That is, this is accomplished, because of showing partial views to the

difference between result of first level base classifiers and result of fusion methods in contrast with result of mediocre base classifiers and result of fusion methods on these experts. Table III reveals these differences.

As shown in Table III in the first row the most excellent individual classifiers are employed in the ensemble network in addition in both of second and third rows the base classifiers are not as well as those are in the first row. Comparison between difference of recognition rate of base classifiers and combining methods exhibits that in utilizing optimized base classifiers the result of fusion methods are not much varies in contrast with using ordinary base classifiers. Moreover when learning rate is 0.025 the result of class indifferent methods are very high compared with result of class conscious methods.

To present how the errors are distributed across the classes confusion matrix is used [20]. Table IV shows the confusion matrix of the recognition results for the most successfully MLP of the mentioned model. For instance, two of the most

misrecognized digits belong to digits 2 and 4 (See Table IV). As shown in Table IV, the network mistakes 66 images of digit 3 for digit 4, and it also mistakes 37 images of digit 3 for digit 2.

## V. CONCLUSION

Combining classifiers to achieve higher accuracy is an important research topic. In this paper, it is tried to improve the prediction efficiency by using ensemble methods. In particular way of using fusion methods, we use of Decision Template and Dempster Shafer methods. Considering the experimental results, the best method in our work is Dempster Shafer method with highest recognition rate of 91.98%. This illustration was given to demonstrate that DT and DS are a richer combiner than the class-conscious combiners.

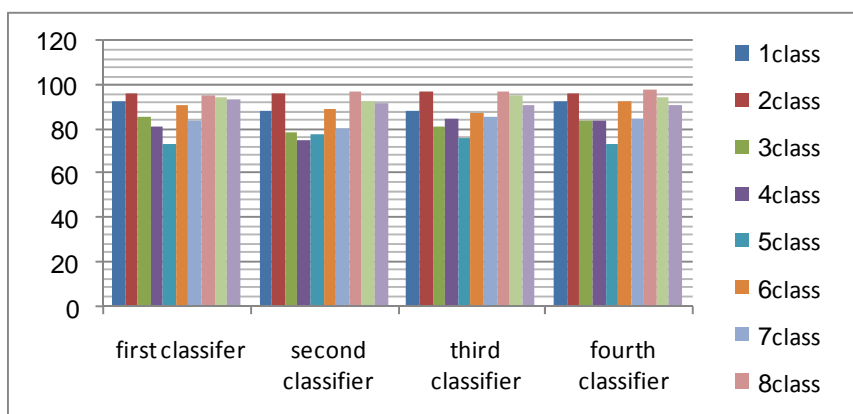


Fig. 3 Recognition rates, averaged over ten test runs, each expert trained with different random initial weights, and different topologies on unseen synthesized images of training set broken down by ten classes

TABLE II  
RECOGNITION RATES (%) OF DIFFERENT FUSION METHODS

Fusion Method	Class Indifferent		Class Conscious				
	DT	DS	MIN	MAX	Sum	Average	Product
Learning Rate=0.001	91.80	<b><u>91.98</u></b>	90.60	90.50	91.28	91.28	<u>91.61</u>
Learning Rate=0.025	86.87	<b><u>87.07</u></b>	81.94	80.14	81.05	81.05	<u>82.70</u>
Learning Rate=0.05	80.58	<b><u>80.73</u></b>	75.49	72.22	73.23	73.23	<u>75.92</u>

In each row various learning rate for base classifiers is applied. The highest recognition rate of each row is typed in bold. And maximum result in class indifferent and class conscious methods are underlined. Each result is the average of ten times testing the corresponding model, each time base classifiers are trained with different random initial weights and different topologies.

TABLE III  
RECOGNITION RATE OF BASE CLASSIFIERS BESIDE THE BEST RESULT OF FUSION METHODS

Recognition Rate of Base Classifiers					Best result of fusion methods (%)	
	Classifier 1 (35 hidden neurons)	Classifier 2 (40hidden neurons)	Classifier 3 (45 hidden neurons)	Classifier 4 (50 hidden neurons)	Best result of class indifferent methods	Best result of class conscious methods
lr=0.001	88.28,0.75	87.27,1.02	88.22,0.77	88.11,0.62	91.98	91.61
lr=0.025	81.32,0.81	81.54,2.13	80.72,0.45	81.03,1.88	87.07	82.70
lr=0.05	73.07,0.89	73.08,2.42	74.63,1.23	75.38,1.00	80.73	75.92

In each row various learning rate for base classifiers is applied. Values are the average (display only % correct on the test set) and standard deviation of ten times testing the corresponding model, each time base classifiers are trained with different random initial weights and different topologies. Fourfold cross validation exhibits that 600 epoch is sufficient.

## REFERENCES

- [1] Fevzi Alimoglu, Ethem Alpaydin, "Combining Multiple Representations for Pen-based Handwritten Digit Recognition," Turk J.Elec.Eng., VOL.9,NO.1 2001.
- [2] Cheng-Lin Liu, Ching Y. Suen, A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters, Pattern Recognition, 2008.
- [3] S. Haykin, Neural Networks—A Comprehensive Foundation, second ed., Prentice-Hall, 1998.
- [4] K. Woods, W.P. Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell, Vol. 19, 405-410, 1997.
- [5] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their application to handwriting recognition, IEEE Trans. Systems Man Cybernet. Vol. 22, 418-435, 1992.
- [6] K.-C. Ng, B. Abramson, Consensus diagnosis: a simulation study, IEEE Trans. Systems Man Cybernet. Vol.22, 916-928, 1992.
- [7] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. Vol.3, 79-87, 1991.
- [8] L.A. Rastrigin, R.H. Erenstein, Method of Collective Recognition, Energoizdat, Moscow, 1982.
- [9] E. Alpaydin, M.I. Jordan, Local linear perceptrons for classification, IEEE Trans. Neural Networks, Vol. 7, No. 3, 788-792, 1996.
- [10] Hasan Soltanzadeh, Mohammad Rahmati, Recognition of Persian handwritten digits using image profiles of multiple orientations, Pattern Recognition Letters 25, 1569-1576, 2004.
- [11] Turk, M. and Pentland, A.: Eigenfaces for Recognition. J. Cognitive Neurosci. VOL.3,NO. 1, 71-86,1991.
- [12] Martinez A, Kak A, PCA versus LDA. IEEE Trans Pattern Anal Mach Intell VOL.23, No.2, 228-233,2001.
- [13] Nadal, C., R. Legault and C.Y. Suen, "Complementary Algorithms for Recognition of totally Unconstrained Handwritten Numerals, " Proc. 10th Int. Conf. Pattern Recognition, Vol. A, pp. 434-449, 1990.
- [14] A. Al-Ani, and M. Deriche, "A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence," Journal of Artificial Intelligence Research, vol. 17, pp. 333-361, 2002.
- [15] L. I. Kuncheva, James C. Bezdek, Robert P.W. Duin, "Decision Templates for Multiple Classifier Fusion: An Experimental Comparison," Pattern Recognition, vol. 34, no.2, pp. 299-314, 2001.
- [16] L. I. Kuncheva, R.K. Kounchev and R.Z. Zlatev, "Aggregation of multiple classification decisions by fuzzy templates," Third European Congress on Intelligent Technologies and Soft Computing, EUFIT'95, pp. 1470-1474, 1995.
- [17] L.I. Kuncheva, "Combining Pattern Classifiers: Methods and algorithms," published by John Wiley & Sons. Inc., 2004.
- [18] G. Rogova, "Combining the results of several neural network classifiers," Neural Networks, vol. 7, pp. 777-781, 1994.
- [19] C.Y. Suen, S. Izadnia, J. Sadri, F. Solimanpour, Farsi script recognition: a survey, in: Proceedings of the Summit on Arabic and Chinese Handwriting Recognition, University of Maryland, College Park, MD, 2006, pp. 101-110.
- [20] Catherine A. Shipp, Ludmila I. Kuncheva, "Relationship between Combination methods and measures of diversity in combining classifiers", Information Fusion, Vol.3, pp.135-148, 2002.
- [21] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," Connect. Sci. 8, pp. 385-404, 1996.
- [22] C. Dietrich, G. Palm, and F. Schwenker. Decision templates for the classification of bioacoustic time series. Information Fusion, 4:101-109, 2003.
- [23] C. Dietrich, F. Schwenker, and G. Palm. Classification of time series utilizing temporal and decision fusion. In J. Kittler and F. Roli, editors, Proc. Second International Workshop on Multiple Classifier Systems, volume 2096 of Lecture Notes in Computer Science, Cambridge, UK, 2001, Springer-Verlag, pp. 378-387.
- [24] J. Kittler, M. Balette, J. Czyz, F. Roli, and L. Vanderdorpe. Decision level fusion of intramodal personal identity verification experts. In F. Roli and J. Kittler, editors, Proc. 2nd International Workshop on Multiple Classifier Systems, Vol. 2364 of Lecture Notes in Computer Science, Cagliari, Italy, Springer-Verlag, pp. 314-324,2002.
- [25] G. Giacinto, F. Roli, and L. Didaci. Fusion of multiple classifier for intrusion detection in computer networks. Pattern Recognition Letters, 24:1795-1803, 2003.
- [26] L. I. Kuncheva. "Fuzzy" vs "non-fuzzy" in combining classifiers designed by boosting. IEEE Transactions on Fuzzy Systems 11:729-741, 2003.
- [27] L. I. Kuncheva. Using measures of similarity and inclusion for multiple classifier fusion by decision templates. Fuzzy Sets and Systems, 122(3):401-407, 2001.
- [28] Y. Lu. Knowledge integration in a multiple classifier system. Applied Intelligence, 6:75-86, 1996.
- [29] A. Goltsev, D.Rachkovskij, Combination of the assembly neural network with a perceptron for recognition of handwritten digits arranged in numeral strings, Pattern Recognition 38, 315 - 322, 2005.
- [30] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, 226-239,1998.
- [31] A. F. R. Rahman, M. C. Fairhurst, Multiple classifier decision combination strategies for character recognition: A review, International Journal On Document Analysis And Recognition, 166-194, 2003.
- [32] C. L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition 36, 2271 - 2285, 2003.
- [33] O.D. Trier, A.K. Jain, T. Taxt, Feature extraction methods for character recognition—a survey, Pattern Recognition Vol.29, No. 4,641-662,1996.

- [34] Ho TK, Hull JJ, Srihari SN (1994) Decision combination in multiple classifier systems. *IEEE Trans Pattern Anal Mach Intell* 16(1):66–75,1994.
- [35] Xu L, Krzyzak A, Suen CY, Associative switch for combining multiple classifiers. In: *Int. Joint Conf. on Neural Networks*, vol.1. pp 43–48, 1991.
- [36] Suen CY, Nadal C, Mai TA, Legault R, Lam L, Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. In: *Proc.IWFHR*, pp 131-143 ,1990.
- [37] A. Amin, Off-line Arabic character recognition: the state of the art, *Pattern Recognition* 31 517–530, 1998.

**Reza Ebrahimpour** was born in Mahallat, Iran, in July 1977. He received the BS degree in electronics engineering from Mazandaran University, Mazandaran, Iran and the MS degree in biomedical engineering from Tarbiat Modarres University, Tehran, Iran, in 1999 and 2001, respectively. He received his PhD degree in July 2007 from the School of Cognitive Science, Institute for Studies on Theoretical Physics and Mathematics, where he worked on view-independent face recognition with Mixture of Experts. His research interests include human and machine vision, neural networks, pattern recognition.

**Fatemeh Sharifzadeh** received the B.Sc. degree in Computer Sciences from Tehran University, Tehran Iran, in 2009. Her research interests include human and machine vision, neural networks, pattern recognition.