Improvement of face recognition performance using a new hybrid subspace classifier

Serkan Keser (skeser@ahievran.edu.tr)
Kırşehir Ahi Evran University

Research Article

Keywords: DCVA, Multiple classification systems, Fisherface, Hybrid DCVA-Fisherface

Posted Date: October 10th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2125344/v1

Improvement of face recognition performance using a new hybrid subspace classifier

***Serkan Keser[1]**

[1] Kırşehir Ahi Evran University, Department of Electrical-Electronics Engineering, Bağbaşı Mah. Sahir Kurutluoğlu Cad. No: 100, Kırşehir, TURKEY

skeser@ahievran.edu.tr

Tel: 0 386 280 60 70

Fax: 0 386 280 38 01

ORCID: 0000-0001-8435-0507

ABSTRACT

Multiple Classification Systems (MCS) play an important role in increasing recognition performance, especially when using heterogeneous classifiers effectively improves performance. In this study, a new hybrid classifier was designed using heterogeneous Fisherface and Discriminative common vector approach (DCVA) subspace recognition methods, which gave successful results in face recognition. The main feature of the hybrid classifier is that it can assign the misclassified parts to the correct class. While DCVA is based on the common properties of the signals belonging to the classes for classification, Fisherface is based on the different properties of the signals. In order to create a hybrid classifier, called the Hybrid DCVA-Fisherface (HDF), the classifiers' decision rules were combined using the Minimum Proportional Score Algorithm (MPSA) and Recognition Update Algorithm (RUA). To better examine the efficiency of the algorithms, tests were also carried out by downsampling the images. When the experimental results were analyzed, the proposed hybrid classifier gave higher recognition rates than other classifiers.

Keywords DCVA · Multiple classification systems · Fisherface · Hybrid DCVA-Fisherface

1 Introduction

Traditionally, classifiers are used separately for classification in pattern recognition, and the classifier that gives the best recognition rate is determined. In addition, classification can be made with multiple classifier groups [1]. Multiple classifier systems (MCS) are presented as high-performance methods in pattern recognition. The MCS method is constructed using combinations of a number of different classifiers. The fact that the classifiers selected by this method are created in the true combination and make different errors affect the classification performance. Methods aiming to design pattern classification systems formed from diverse classifiers as hybrids are accepted as a basic need nowadays. As a result, the MCS method is promising to increase classification performance.

1.1 Related work

Many studies in the literature consist of hybrid classifiers for pattern recognition [2-9]. These studies are listed below. Giacinto Role(2001) first created a classifier ensemble and then proposed an MCS method that selects the most accurate classifiers with the method they used. Mukkamala et al. (2005) used five different classifiers (SVM, MARS, ANN(RP), ANN(SCG), ANN(OSS)) and combined the decision rules of the classifiers with majority voting. Gu and Jin (2012) created a heterogeneous ensemble of classifiers for binary classification of EEG signals with linear discriminant analysis (LDA), Linear support vector machines (L-SVM), and radial basis function-support vector machines (RBF-SVM). Cheng and Chen (2014) implemented the MCS method for face recognition using five classifiers (PCA, Fisherface, spectral regression dimension analysis (SRDA), SLDA, SLPP). Govindarajan and Chandsekaran (2012) proposed an ensemble method that implements a generalized version of the bagging and boosting algorithms that combine the decisions of various classifiers. Partridge and yates use two different performance variants to select M out of C classifiers. The choice of the classifier depends on the recognition performance of the classifiers. Jian Xun Mi et al. proposed a nearest-farthest subspace (NFS) classifier that takes advantage of different properties of the class-specific subspace using the nearest subspace (NS) and farthest subspace FS classifiers. Rodriguez, J. J. et al. proposed the Rotation Forest method to construct ensembles of classifiers based on feature extraction. In this study, they randomly divided the feature set into subset K and applied Principal Component Analysis (PCA) to each subset to generate the training data for the base classifier.

1.2 Motivation and contribution

A general face recognition system includes feature extraction [10,11], classifier selection [12,13] and the classification rule [14]. Principal component analysis (PCA) [15,16] is one of the most common methods used for feature extraction. PCA transforms the original images from a high-dimensional space to a low-dimensional feature space to extract features. Fisher's linear discriminant analysis (FLDA) is one of the most common classifiers used in face recognition [17-18]. FLDA finds orthogonal optimum basis vectors that maximize the within-class distribution and minimize the between-class distribution [19]. In other words, the optimum basis vectors are found using the difference subspace of the matrix formed by the product of the between-class distribution matrix and the inverse of the within-class distribution matrix. The Fisherface method is performed from FLDA and makes the within-class distribution matrix ($\mathbf{S_W}$) non-singular using PCA; in this way, optimal basis vectors can be found [19].

Another classifier used in face recognition that gives good results is the discriminative common vector approach (DCVA) [20-24], a classifier derived from CVA. The CVA uses the indifference subspace of the within-class distribution matrix ($\mathbf{S_W}$). Using CVA, a unique vector is found that contains the common properties of each class. This vector is called the common vector, and the dimension of the common vector is equal to the dimension of the samples.

On the other hand, DCVA performs classification using the basis vectors that maximize the distributions of these common vectors. Discriminative common vectors whose sizes are one less than the number of classes are found using DCVA for each class. As a result, Fisherface is classifier based on difference subspace, and DCVA is classifier based on indifference subspace. During classification, when one of these two classifiers misclassifies a test image according to the different (or common properties), the other classifier can classify it correctly according to the common properties (or different properties). Also, even if two classifiers assign a test image to the correct class, the classifiers may have different recognition performances. Based on these ideas, a hybrid classifier was developed using heterogeneous Fisherface and DCVA classifiers, combining the decision rule of the two classifiers and reducing the overall error rate of the system.

Classical classifiers project to test and train images into a subspace using basis vectors and compare them based on the Euclidean distance. The proposed method uses a performance score value instead of Euclidean distances. The algorithm that finds these score values is called the Minimum Proportional Score Algorithm (MPSA). This algorithm finds a score value for each test image for the classifiers. The base classifier is selected according to the performance score of the classifiers. The recognition rate found using Euclidean distances of the chosen base classifier is used in the MPSA. The performance values of the faulty parts found using the base classifier and the performance values of the other class are updated using the Recognition Update Algorithm (RUA). If some of the incorrectly classified parts of the base classifier are assigned to the correct class as a result of the update, the recognition rate increases.

Two separate databases were used in experimental studies. In the test phase, *leave-one-out* cross-validation was used. It has been observed that the HDF method with the used decision rule gives higher recognition rates between 0.25% and 2% compared to the Fisherface and DCVA.

2. The classifiers used in the study

The used DCVA, Fisherface, and HDF classifiers are briefly summarized below.

2.1 DCVA classifier

The DCVA method primarily involves finding the common vectors ($\boldsymbol{x}^i_{com}$) for the *ith* class [20]. Then the common scatter matrix ($\mathbf{S}_{\boldsymbol{com}}$) is found using common vectors as follows,

$$\mathbf{S}_{\boldsymbol{com}} = \sum_{i=1}^{C} (\boldsymbol{x}^i_{com} - \boldsymbol{\mu}_{com})\,(\boldsymbol{x}^i_{com} - \boldsymbol{\mu}_{com})^T, i = 1, \dots, C \tag{1}$$

where $\boldsymbol{\mu}_{com}$ indicates the mean vector of the common vectors. The eigenvectors corresponding to the nonzero eigenvalues of matrix $\mathbf{S_{com}}$ give the optimal projection vectors ($\mathbf{W_{opt}}$=[$\mathbf{w_1}$ $\mathbf{w_2}$… $\mathbf{w_{C-1}}$]) for the DCVA. The feature vectors can be written as follows [20];

$$\boldsymbol{\Omega_i} = [<\mathbf{x^i_r}, \mathbf{w_1}> \cdots <\mathbf{x^i_r}, \mathbf{w_{C-1}}>] \tag{2}$$

These vectors ($\boldsymbol{\Omega_i}$) are called discriminative common vectors, whose dimensions are at most *C-1*. In the test phase, to classify the test signal, the feature vectors of the test signal are found by

$$\boldsymbol{\Omega_{test}} = \mathbf{W^T_{opt}}\boldsymbol{x_{test}} \tag{3}$$

where $\boldsymbol{\Omega_{test}} \in \mathbf{R}^{(C-1)\times 1}$). The operations described above were performed for the insufficient data case (M<n); however, in the sufficient data case (M>n), because the covariance matrix has *n* nonzero eigenvalues, difference and indifference subspaces can be determined by estimation [23].

2.2. Fisherface Classifier

In Fisherface, $\mathbf{S_w}$ and $\mathbf{S_b}$ scattering matrices are found for the image in the training set. The between-class scatter matrix $\mathbf{S_B}$ is calculated by

$$\mathbf{S_B} = \sum_{i=1}^{C} N(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \tag{4}$$

where *N* is the number of samples in a class. $\boldsymbol{\mu_i}$ is the mean of the *i*th class, and $\boldsymbol{\mu}$ represents the mean of all classes; C is the number of classes, and the optimal set of basis vectors ($\mathbf{W_{opt}}$) is determined using these matrices [19].

$$\mathbf{W_{opt}} = \underset{\mathbf{W}}{\mathbf{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|} \tag{5}$$

C-1 eigenvectors corresponding to the largest eigenvalues of the formed matrix as a result of $\mathbf{S_W^{-1}S_B}$ multiplication gives the optimal basis vector ($\mathbf{W_{opt}}$). In other words, these basis vectors are obtained from the difference subspace of $\mathbf{S_W^{-1}S_B}$. However, $\mathbf{S_W}$ becomes singular if the number of images in the database is less than *N* size of images. To solve this problem, all image signals in the database are reduced to *N-c* by applying PCA. Thus, the new $\mathbf{S_W}$ matrix of PCA applied signals becomes non-singular, then applying the standard FLD defined by (5) to reduce the dimension to $C-1$. All the signals in the training set are projected onto the optimum space using this basis vector.

$$\boldsymbol{\Omega}_i = \mathbf{W}_{\mathbf{opt}}^T \mathbf{X}_\mathbf{i}, \quad i = 1,2,\dots,C \tag{7}$$

where $\boldsymbol{\Omega}_i \in \mathbf{R}^{(\mathbf{C}-\mathbf{1})\times\mathbf{K}}$ and, K is the number of samples in the *i*th class. For classification, the test signal is first projected using the $\mathbf{W_{opt}}$, and, then $\boldsymbol{\Omega}_{test}$ ($\boldsymbol{\Omega}_{test} \in \mathbf{R}^{(\mathbf{C}-\mathbf{1})\times\mathbf{1}}$) is found by,

$$\boldsymbol{\Omega}_{test} = \mathbf{W}_{\mathbf{opt}}^T \boldsymbol{x}_{test} \tag{8}$$

In the test phase, the projected test signal ($\boldsymbol{\Omega}_{test}$) is assigned by Euclidean distance measure to the most appropriate class.

2.3. Hybrid DCVA- Fisherface (HDF) Classifier

In the study, a hybrid classifier was performed by combining the decision rules of DCVA and Fisherface. In the test phase, the recognition matrices (*euc_class*^{1},*euc_class*^{2}) are found using the Euclidean distance measure of all test signals for each classifier. These matrices are essential to determine whether the test signals are assigned to the correct class. Two different algorithms have been proposed to obtain hybrid classifiers. In the first algorithm, MPSA, performance score values based on the recognition performances of the classifiers were found instead of the Euclidean distance criterion used by the classifiers. To achieve this, the classification matrices (*euc_class*^{1},*euc_class*^{2}) are converted into performance score matrices (SC_{ij}^1, SC_{ij}^2). By using the MPSA algorithm, which classifier will be taken as the base classifier is determined. After the base classifier is selected, the parts classified incorrectly by the base classifier are updated according to the performance scores of the other classifier with the Recognition Update Algorithm (RUA) algorithm. The recognition rate will increase if some errors are assigned to the correct classes. However, if no errors are corrected, the recognition rate of the hybrid classifier becomes equal to the recognition rate of the selected base classifier. The proposed hybrid classifier system is shown in Fig.1 below.

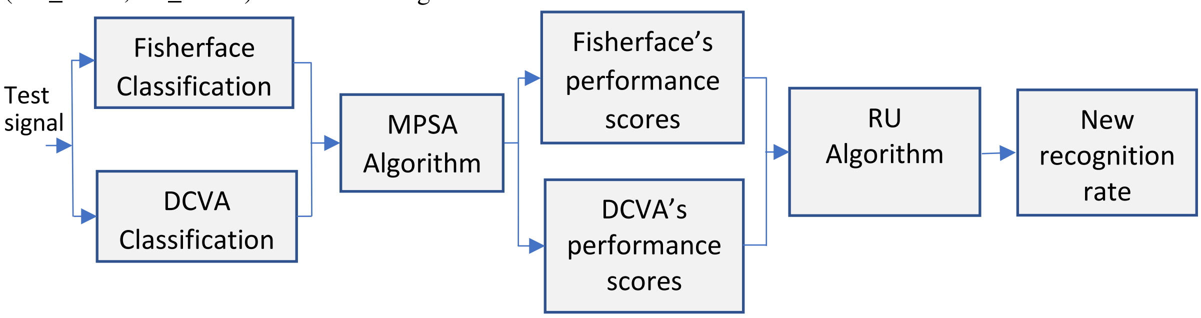
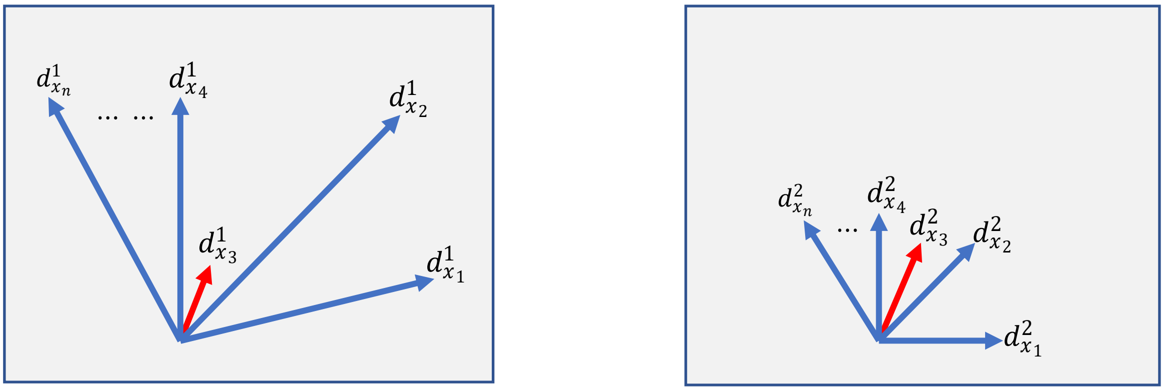


Fig. 1 Block diagram of the proposed HDF classifier

2.3.1. Minimum Proportional Score Algorithm (MPSA)

Classically, the class given the smallest Euclidean value is assigned a test signal using the classifiers, but the Euclidean distances of the other classes are ignored. On the other hand, Euclidean distances of other classes also give information about the performance of the classifier. The sum of the ratios between the class giving the smallest Euclidean value and the Euclidean distances of the other classes can provide a performance score value for the test signals. Based on this idea, the performance score was used for classification with the proposed MPSA. In the MPSA, the classification is made according to the obtained performance score values instead of classifying classifiers according to the Euclidean distance criterion. In addition, the base classifier that gives the highest performance of the two classifiers is determined. The test signals that the base classifier classifies incorrectly according to the Euclidean distance criterion are tried to be assigned to the correct class by using the performance scores of the other classifier and the base classifier. In Fig.2 below, a test signal is projected into the optimum subspace separately for two classifiers, and the Euclidean distances obtained for *n* classes are shown as vectors according to their magnitudes.



a) **b)**

Fig. 2 Euclidean distance measures of *classifiers*^{1} (a) and *classifiers*^{2} (b) for a test signal

The distances indicated by the red arrow represent the assignment to the correct class. Other arrows indicate longer Euclidean distances belonging to different classes. For example, they are the magnitudes of the distances belonging to the faulty classes. Although *classifiers*^{1} and *classifiers*^{2} assign the same correct class, they have a recognition performance difference. However, as seen in Fig. 2, while there is a high distance difference between the correct and false classes in *classifiers*^{1}. This difference is much less for classifiers^2. In other words, it can be said that classifiers^1 gives a better recognition performance than *classifiers*^{2}. With the proposed mathematical method, the performance score matrix ($\mathbf{SC}_{ij}^p$) of the classifiers for samples belonging to all classes is as follows.

$$\mathbf{SC}_{ij}^p = \left(\frac{d_{x_{min}}^p}{d_{x_{ij}}^p}\right)^2 + \left(\frac{d_{x_{min}}^p}{d_{x_{ij}}^p}\right)^2 \ldots + \left(\frac{d_{x_{min}}^p}{d_{x_{ij}}^p}\right)^2, \tag{9}$$

i=1,2,…,n and j=1,2,…,k,

where $d_{x_{min}}^p$ is the smallest Euclidean distance between classes for the test signal *x*, *i* is the index of the classes, *n* is the number of classes, *p* is the classifier index, *j* is the index of the test signal, and *k* is the number of samples. The performance values of the classifiers are assigned using the following algorithm. The squaring of the performance score values was used to clarify the difference between the scores. For all test signals, the classifiers have an overall performance score matrix (SC_{ij}^2, SC_{ij}^1). Then the differences between the performance score values for the *jth* test signal of the *ith* class are found ($diff_{i,j} = SC_{ij}^2$-SC_{ij}^1). If this difference is positive, *classifiers*^{1} has better performance. This difference is assigned to the performance value ($perf^1$) of the classifier^1. If the difference is negative, *classifiers*^{2} has better performance. In this case, the performance value ($perf^2$) of the *classifier*^{2} is assigned the absolute value of this difference. At the end of the algorithm, the $perf^1$ and $perf^2$ performance values are summed and assigned to *ts*^{1} and *ts*^{2}, respectively. If the total score 1 (*ts*^{1}) is higher than the total score 2 (*ts*^{2}), it indicates that the first classifier performs better than the second classifier for all test signals. As a result, the first classifier is selected; otherwise, the second classifier is selected. The MPS algorithm is given below.

Algorithm 1. Minimum Proportional Score Algorithm (MPSA)

Input: SC_{ij}^1, SC_{ij}^2

Output: $perf^1$, $perf^2$, ts^1, ts^2, $b\_classifier$ //b_classifier: base classifier.

for $i = 1{:}\,n$ **do**
  for $j = 1{:}\,k$ **do**
    $diff_{i,j} = SC_{ij}^2$-SC_{ij}^1
    if $diff_{i,j} > 0$
      $s1 = s1 + 1$
      $perf^1(s1) = diff_{i,j}$
    else
      $s2 = s2 + 1$
      $perf^2(s2) = |diff_{i,j}|$
    end for
  end for
end for
$ts^1 = sumation(perf^1)$
$ts^2 = sumation(perf^2)$
if $ts^1 > ts^2$
$b\_classifier = classifier^1$
else
$b\_classifier = classifier^2$
end for
end algorithm

The base classifier is selected using the MPSA, and then the RUA is applied. The *euc_class1* mentioned above and *euc_class2* matrices are converted to *class*^{1} and *class*^{2} matrices containing the values cc when classified correctly for each test signal and *fc* when classified incorrectly. These matrices are size *n*×*k* for *n* classes and *k* test signals.

2.3.2. Recognition Update Algorithm (RUA)

It is based on the recognition matrix (*euc_class*) obtained according to the Euclidean values of the base classifier using the RUA. In addition, misclassified parts in this matrix are updated according to the performance score values of the other classifier and the base classifier (SC_{ij}^2, SC_{ij}^1). For a test signal, the updated recognition matrix (*updated_rec*) gets the value of "*cc* (correct classification)" if both classifiers correctly classify, whereas the updated recognition matrix receives the value of "*fc* (false classification)" if they classify incorrectly. The *updated_rec* matrix is size *n*×*k* for n classes and k test images. The RUA algorithm is given below.

Algorithm 2. Recognition Update Algorithm (RUA)

Input: $SC_{ij}^2, SC_{ij}^1, class^1, class^2, b\_classifier$

Output: $updated\_rec$

for $i = 1{:}\,n$ **do**
  for $j = 1{:}\,k$ **do**
    if $(class^1(i,j) = cc)$ & $(class^2(i,j) = cc)$
     $updated\_rec\{i,j\} = cc;$
     elseif $(class^1(i,j) = fc)$ & $(class^2(i,j) = fc)$
     $updated\_rec\{i,j\} = fc;$
     elseif $(class^1(i,j) = cc)$ & $(class^2(i,j) = fc)$& $SC_{ij}^1 < SC_{ij}^2$ & $b\_classifier = classifier^2$
     $updated\_rec\{i,j\} = cc;$
     elseif $(class^1(i,j) = fc)$ & $(class^2(i,j) = cc)$& $SC_{ij}^1 > SC_{ij}^2$ & $b\_classifier = classifier^1$
     $updated\_rec\{i,j\} = cc;$
     elseif $(class^1(i,j) = cc)$ & $(class^2(i,j) = fc)$& $SC_{ij}^2 < SC_{ij}^1$& $b\_classifier = classifier^2$
     $updated\_rec\{i,j\} = fc;$
     elseif $(class^1(i,j) = fc)$ & $(class^2(i,j) = cc)$& $SC_{ij}^2 > SC_{ij}^1$& $b\_classifier = classifier^1$
     $updated\_rec\{i,j\} = fc;$
     elseif $(class^1(i,j) = fc)$ & $(class^2(i,j) = cc)$ & $b\_classifier = classifier^2$
     $updated\_rec\{i,j\} = cc;$
     elseif $(class^1(i,j) = cc)$ & $(class^2(i,j) = fc)$ & $b\_classifier = classifier^1$
     $updated\_rec\{i,j\} = cc;$
     end for
  end for
end for
end algorithm

To indicate the RUA more clearly, the *class¹* and *class²* recognition matrices of the two classifiers are given in Fig. 3 below. The first classifier (*classifier¹*) is assumed to be selected as the base classifier using the MPS algorithm. In this case, only the recognition matrix (*class¹*) of *classifier¹* is referenced.

cc	cc	cc	cc	cc	cc
cc	cc	cc	cc	cc	cc
cc	cc	cc	fc	cc	cc
cc	cc	cc	cc	cc	fc $s^1_{4,6}=6$
cc	cc	cc	cc	fc $s^1_{5,5}=7$	cc

a) Base classifier matrix (*class¹*)

cc	fc	cc	fc	cc	cc
cc	cc	cc	cc	cc	fc
cc	cc	cc	fc	fc	cc
cc	cc	cc	cc	cc	cc $s^2_{4,6}=4$
cc	cc	cc	cc	cc $s^2_{5,5}=3$	cc

b) Other classifier matrix (*class²*)

Fig. 3 Base classifier (a) and Other classifier (b) matrices

As you can see, there are three errors in the *class*^{1} matrix. First, blue boxes in Fig. 3 (a) and (b) indicate that both classifiers misclassify for the same test signal. Therefore, this error cannot be assigned to the correct class. However, when the two errors in the *class*^{1} matrix of the base classifier are compared with the corresponding recognition values (*fc* or *cc*) and scores in the *class*^{2} matrix, it is seen that *classifier*^{2} has better performance scores than *classifier*^{1} ($s^2_{4,6}<s^1_{4,6}$, $s^2_{5,5}<s^1_{5,5}$). As a result, these misclassified parts are assigned to the correct classes using the RU algorithm, and the updated recognition matrix is obtained in Fig. 4 below.

cc	cc	cc	cc	cc	cc
cc	cc	cc	cc	cc	cc
cc	cc	cc	fc	cc	cc
cc	cc	cc	cc	cc	cc
cc	cc	cc	cc	cc	cc

Fig. 4 The updated recognition matrix

3. Experimental results

ORL and YALE face databases were used in the study. The ORL database is a face database consisting of 400 images obtained using ten different images of 40 people. The YALE database has a total of 165 images belonging to 15 people. The study, 150 images were used because two images of each person were similar. Images are downsampled to create sufficient data cases. Downsampling was performed using two factors, and one of the four obtained images was used in the studies. This selected image has been downsampled similarly, and its size has been further reduced. By using the downsampling process, 64×64,32×32,16×16 and 8×8 sized images for ORL and 80×80,40×40,20×20 and 10×10 sized images were used for YALE. Experimental results were found using codes written in a MATLAB environment. The results for ORL and YALE databases are given in Table 1 and Table 2 below, respectively. The + and – symbols next to the expressions showing the image dimensions indicate the sufficient and insufficient data cases, respectively.

Tablo 1. ORL database için bulunan tanıma oranları

	Image sizes			
Classifiers	**$64\times64^-$**	**$32\times32^-$**	**$16\times16^+$**	**$8\times8^+$**
DCVA	99.5	99.25	99	97.75
Fisherface	99	99	97	99.25
HDF	**99.75**	**99.5**	**99.5**	**99.5**

Tablo 2. YALE database için bulunan tanıma oranları

	Image sizes			
Classifiers	**$80\times80^-$**	**$40\times40^-$**	**$20\times20^-$**	**$10\times10^+$**
DCVA	98.34	97.34	97.34	93.34
Fisherface	98.67	96.67	98.67	98.34
HDF	**99.67**	**99.34**	**99.67**	**99.34**

Tables 1 and 2 show that the Hybrid-DF classifier has higher recognition rates than Fisherface and DCVA classifiers. The highest difference between the recognition rates of HDF and the others was 2%, which was found using 40×40 images for the YALE database. For the ORL database, the highest difference is 0.5%. In these tables, the recognition rates for HDF are higher than those of other classifiers because the MPSA correctly chooses the base classifier. Therefore, in the following Table 3, the selected base classifiers using the MPSA algorithm are given for all dimensions of the images.

Tablo 3. *The selected base classifiers using the MPSA*

YALE DATABASE		*ORL DATA BASE*	
Image sizes	Choosen Classifiers	Image sizes	Choosen Classifiers
80×80	Fisherface	64×64	DCVA
40×40	DCVA	32×32	DCVA
20×20	Fisherface	16×16	DCVA
10×10	Fisherface	8×8	Fisherface

When the selected classifiers are examined according to the tables above, it can be seen that the classifiers with the highest recognition rate were selected as the base classifier.

4. CONCLUSIONS

Hybrid classifiers have an essential role in pattern recognition. Significantly, using heterogeneous classifiers in hybrid form can give higher recognition rates. For example, the fact that Fisherface and DCVA use different subspaces for classification in the study is a factor that increases the performance of the hybrid classifier. Because a test image classified incorrectly according to the difference subspace can be classified correctly according to the indifference subspace, and vice versa.

In the study, the base classifier with the best performance was selected using the MPSA, and then the recognition matrix was updated using the RUA. The MPSA has chosen a classifier corresponding to the classifier with the highest recognition rate for all test signals as the base classifier for all image sizes. It has been observed in experimental studies that the proposed hybrid classifier gives higher recognition rates than DCVA and Fisherface for all image dimensions.

Declarations

Ethical statement I declare that all the principles of ethical and professional conduct have been followed while preparing the manuscript for publication in signal, image and video processing, and comply with Springer's ethical policies.

Competing interests The author declares no competing interests.

Conflict of interest I declare that no conflicts of interest are relevant to this article's content.

Funding No funding was received to assist with the preparation of this manuscript

Data availability The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

REFERENCES

[1]. Woźniak, M., Grana, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. Information Fusion, 16, 3-17.
[2]. Giacinto, G., & Roli, R. (2001). An approach to the automatic design of multiple classifier systems. Pattern Recognition Letters, 22, 25–33.
[3]. Mukkamala, S., Sung, A. H., & Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. Journal of network and computer applications, 28(2), 167-182.
[4]. Gu S, Jin Y. Heterogeneous classifier ensembles for EEG-based motor imaginary detection. In: 2012 12th UK workshop on computational intelligence (UKCI). IEEE; 2012. p. 1–8.
[5]. Cheng J, Chen L. A weighted regional voting based ensemble of multiple classifiers for face recognition. In: International symposium on visual computing. Springer; 2014. p. 482–91
[6]. Govindarajan M, Chandrasekaran R. Intrusion detection using an ensemble of classification methods. In: World congress on engineering and computer science, vol. 1. 2012. p. 1–6.
[7]. D. Partridge and W. B. Yates, \Engineering multiversion neural-net systems", Neural Computation 8, 869-893 (1996)
[8]. Mi, J. X., Huang, D. S., Wang, B., & Zhu, X. (2013). The nearest-farthest subspace classification for face recognition. Neurocomputing, 113, 241-250.
[9]. Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. IEEE transactions on pattern analysis and machine intelligence, 28(10), 1619-1630.
[10]. Y. Xu, D. Zhang, J. Yang, A. Yang, Two-phase test sample aparse representation method for use with face recognition, IEEE Trans. Circuits Syst. Video Technol. (2011).
[11]. B. Li, C.H. Zheng, D.S. Huang, Locally linear discriminant embedding: an efficient method for face recognition, Pattern Recognition 41 (2008) 3813–3821.
[12]. Z. Fan, Y. Xu, D. Zhang, Local linear discriminant analysis framework using sample neighbors, IEEE Trans. Neural Networks 22 (2011) 1119–1132.
[13]. Y. Pang, Y. Yuan, K. Wang, Learning optimal spatial filters by discriminant analysis for brain–computer-interface, Neurocomputing 77 (2012) 20–27.
[14]. D.S. Huang, J.X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, Ieee Trans. Neural Networks 19 (2008) 2099–2115.
[15]. Ebied, H. M. (2012, May). Feature extraction using PCA and Kernel-PCA for face recognition. In 2012 8th International Conference on Informatics and Systems (INFOS) (pp. MM-72). IEEE.
[16]. Hidayat, E., Fajrian, N. A., Muda, A. K., Huoy, C. Y., & Ahmad, S. (2011, December). A comparative study of feature extraction using PCA and LDA for face recognition. In 2011 7th international conference on information assurance and security (IAS) (pp. 354-359). IEEE.
[17]. Chen, S., Liu, J., & Zhou, Z. H. (2004). Making FLDA applicable to face recognition with one sample per person. Pattern recognition, 37(7), 1553-1555.
[18]. Liu, Q., Lu, H., & Ma, S. (2004). Improving kernel Fisher discriminant analysis for face recognition. IEEE transactions on circuits and systems for video technology, 14(1), 42-49.
[19]. Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence, 19(7), 711-720.
[20]. Cevikalp, H., Neamtu, M., Wilkes, M., & Barkana, A. (2005). Discriminative common vectors for face recognition. IEEE Transactions on pattern analysis and machine intelligence, 27(1), 4-13.
[21]. He, Y. H., Zhao, L., & Zou, C. R. (2005, August). Kernel discriminative common vectors for face recognition. In 2005 International Conference on Machine Learning and Cybernetics (Vol. 8, pp. 4605-4610). IEEE.
[22]. Wen, Y. (2012). An improved discriminative common vectors and support vector machine based face

recognition approach. Expert Systems with Applications, 39(4), 4628-4632.
[23]. Zhu, L., Jiang, Y., & Li, L. (2008, September). Making discriminative common vectors applicable to face recognition with one training image per person. In 2008 IEEE Conference on Cybernetics and Intelligent Systems (pp. 385-387). IEEE.
[24]. Wen, Y. (2017, March). A novel dictionary based SRC for face recognition. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2582-2586). IEEE.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Matlabcodesanddatabases.rar