Signal, Image and Video Processing (2023) 17:2511–2520
https://doi.org/10.1007/s11760-022-02468-w

ORIGINAL PAPER

Improvement of face recognition performance using a new hybrid subspace classifier

Serkan Keser[1]

Received: 2 October 2022 / Revised: 14 December 2022 / Accepted: 22 December 2022 / Published online: 5 January 2023

Abstract

Multiple classification systems play an important role in increasing recognition performance, especially when using heterogeneous classifiers that effectively improve 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. While the classification process of DCVA is based on the common properties of signals belonging to the classes, the classification process of Fisherface is based on the different properties of signals. To create a hybrid classifier, called the Hybrid DCVA-Fisherface, the classifiers' decision rules were combined using the Minimum Proportional Score Algorithm and Recognition Update Algorithm. In addition to the proposed subspace classifiers, convolutional neural networks, Transform learning-Alexnet, Alexnet + SVM, and Alexnet + KNN were used for classification. Studies were conducted using the ORL, YALE, Extended YALE B and Face Research Lab London Set (FRLL). To better examine the efficiency of the algorithms, tests were also carried out by downsampling the images. When the experimental results were analysed, the proposed hybrid classifier gave higher recognition rates than all classifiers for ORL, YALE, and Extended YALE B. However, deep learning methods generally achieved better recognition performance than subspace classifiers for the FRLL database, which has more classes than other databases.

Keywords DCVA · Multiple classification systems · Fisherface · HDF · CNN · Alexnet

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 some different classifiers. The classification performance is affected by the fact that the classifiers selected by the MCS method are created in the correct combination and make different errors. 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 and Roli [2] 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. [3] 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 [4] created a heterogeneous ensemble of classifiers for the 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 [5] implemented the MCS method for face recognition using five classifiers (PCA, Fisherface, spectral regression dimension analysis (SRDA), SLDA, and SLPP). Govindarajan and Chandsekaran [6] proposed an ensemble

✉ Serkan Keser
skeser@ahievran.edu.tr

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

method that implements a generalised version of the bagging and boosting algorithms that combine the decisions of various classifiers. Nweke et al. [7] implemented a different data fusion and multiple classifier systems in human activity recognition. Mi et al. [8] 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 et al. [9] used 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. A multi-feature, multi-classifier system for SER is proposed in [10], and four convolutional neural networks (CNNs) and a traditional support vector machine (SVM) classifier are used to exploit emotional information in multiple features.

Apart from these studies, there are new face recognition studies that give successful results in computer vision. For example, Liao and Gu [11] proposed a face recognition approach by subspace extended sparse representation and discriminative feature learning, called SESRC & LDF. The experimental results show that SESRC & LDF achieves the highest recognition rates, outperforming many algorithms. In another study, they also performed a new subspace clustering method based on alignment and graph embedding (SCAGE) [12]. In SCAGE, they unify the image alignment process and clustering subspace learning process based on low rank and sparse representation. Liao et al. [13] created a graph-based adaptive and discriminative subspace learning method (GADSL), giving successful face image clustering results.

1.2 Motivation and contribution

A general face recognition system includes feature extraction [14, 15], classifier selection [16, 17], and the classification rule [18]. Principal component analysis (PCA) [19, 20] is one of the most common methods used for feature extraction. The 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 [21, 22]. FLDA finds orthogonal optimum basis vectors that maximise the within-class distribution and minimise the between-class distribution [23]. 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 [23].

Another classifier used in face recognition that gives good results is the Discriminative Common Vector Approach (DCVA) derived from CVA [24–28]. The CVA uses the indifference subspace of the within-class distribution matrix ($\mathbf{S_W}$), and a unique vector containing each class's common properties is found using CVA. 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 maximise 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 a classifier based on difference subspace, and DCVA is a classifier based on indifference subspace. During classification, when one of these 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 performance score values 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 belonging to classifiers for each test image. The base classifier is selected according to the performance score of the classifiers. The recognition rate found using Euclidean distances of the classifiers is used in the MPSA. The base and another classifier's performance values are obtained to correct incorrectly classified parts. The algorithm called the Recognition Update Algorithm (RUA) performs this correction. Due to the update, the recognition rate increases if some incorrectly classified parts of the base classifier are assigned to the correct class.

Four separate databases, ORL, YALE, Extended YALE B, and FRLL, were used in experimental studies. In the test phase, *leave-one-out* cross-validation was used for YALE and ORL, threefold cross-validation was performed for Extended YALE B, and tenfold cross-validation was carried out for the FRLL database. It has been observed that the HDF method with the used decision rule gives higher recognition rates between 0.25% and 20% than the Fisherface and DCVA. In addition, the face recognition performances of deep learning methods such as CNN, Alexnet + SVM, Alexnet + KNN, and Transfer Learning- Alexnet (TrAlexnet) and subspace classifiers were also compared.

2 The classifiers used in the study

DCVA, Fisherface, HDF, CNN, Alexnet + SVM, Alexnet + KNN, and TrAlexnet classifiers used in this study are briefly summarised below.

2.1 DCVA classifier

The DCVA method primarily involves finding the common vectors ($\boldsymbol{x}_{com}^{i}$) for the *ith* class [24]. Then the common scatter matrix ($\mathbf{S}_{com}$) is found using common vectors as follows,

$$S_{com} = \sum_{i=1}^{C}\left(x_{com}^{i} - \mu_{com}\right)\left(x_{com}^{i} - \mu_{com}\right)^{T} \tag{1}$$

where $\boldsymbol{\mu}_{com}$ indicates the mean vector of the common vectors, and *i* is the index of classes. The eigenvectors corresponding to the nonzero eigenvalues of matrix $\mathbf{S}_{\mathbf{com}}$ give the optimal projection vectors ($\mathbf{W}_{\mathbf{opt}}$=[$\mathbf{w}_{\mathbf{1}}\mathbf{w}_{\mathbf{2}}\ldots\ \mathbf{w}_{\mathbf{C\text{-}1}}$]) for the DCVA. The feature vectors can be written as follows [24];

$$\Omega_{i} = \left[< \mathrm{x}_{\mathrm{r}}^{\mathrm{i}}, \mathrm{w}_{1} > \ldots < \mathrm{x}_{\mathrm{r}}^{\mathrm{i}}, \mathrm{w}_{\mathrm{C}-1} >\right] \tag{2}$$

These vectors (Ω_{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

$$\Omega_{test} = \mathrm{W}_{\mathrm{opt}}^{\mathrm{T}} x_{test} \tag{3}$$

where $\Omega_{test} \in \mathrm{R}^{(\mathrm{C}-1)\times 1}$. 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 [27].

2.2 Fisherface classifier

In Fisherface, first, $\mathbf{S}_{\mathbf{w}}$ and $\mathbf{S}_{\mathbf{b}}$ scattering matrices are obtained for the image in the training set. The between-class scatter matrix $\mathbf{S}_{\mathbf{B}}$ is calculated by

$$\mathrm{S}_{\mathrm{B}} = \sum_{i=1}^{C} N(\mu_{i} - \mu)(\mu_{i} - \mu)^{T} \tag{4}$$

where *N* is the number of samples in a class. $\boldsymbol{\mu}_{\boldsymbol{i}}$ 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 ($\mathbf{W}_{\mathbf{opt}}$) is determined using these matrices [23].

$$\mathrm{W}_{\mathrm{opt}} = \underset{\mathrm{W}}{\mathrm{argmax}} \frac{\left|W^{T} S_{B} W\right|}{\left|W^{T} S_{W} W\right|} \tag{5}$$

C-1 eigenvectors corresponding to the largest eigenvalues of the formed matrix as a result of $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$ multiplication gives the optimal basis vector ($\mathbf{W}_{\mathbf{opt}}$). In other words, these basis vectors are obtained from the difference subspace of $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$. However, $\mathbf{S}_{\mathbf{W}}$ becomes singular if the number of images in the database is less than the *N* size of images. All image signals in the database are reduced to *N-c* by applying PCA to solve this problem. Thus, the new $\mathbf{S}_{\mathbf{W}}$ matrix of PCA applied signals becomes non-singular, 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.

$$\Omega_{i} = \mathrm{W}_{\mathrm{opt}}^{T}\mathrm{X}_{\mathrm{i}},\ i = 1, 2, \ldots, C \tag{6}$$

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

$$\Omega_{test} = \mathrm{W}_{\mathrm{opt}}^{T} x_{test} \tag{7}$$

In the test phase, the projected test signal ($\boldsymbol{\Omega}_{\boldsymbol{test}}$) 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 for the classification. 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 to be taken as the base classifier is determined. After the base classifier is selected, the incorrectly classified parts by the base classifier are updated using the RUA according to the performance scores of the other classifier. The recognition rate increases 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.

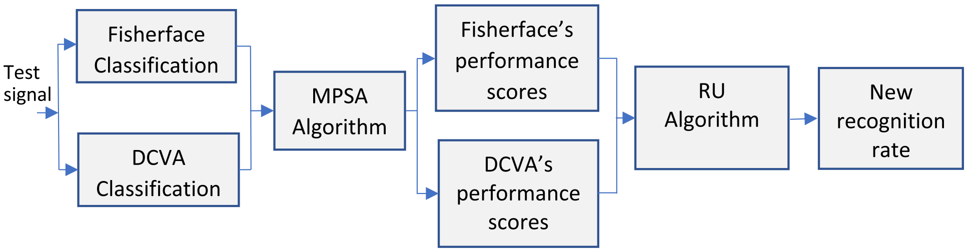

Test signal
Fisherface Classification
DCVA Classification
MPSA Algorithm
Fisherface's performance scores
DCVA's performance scores
RU Algorithm
New recognition rate

Fig. 1 Block diagram of the proposed HDF classifier

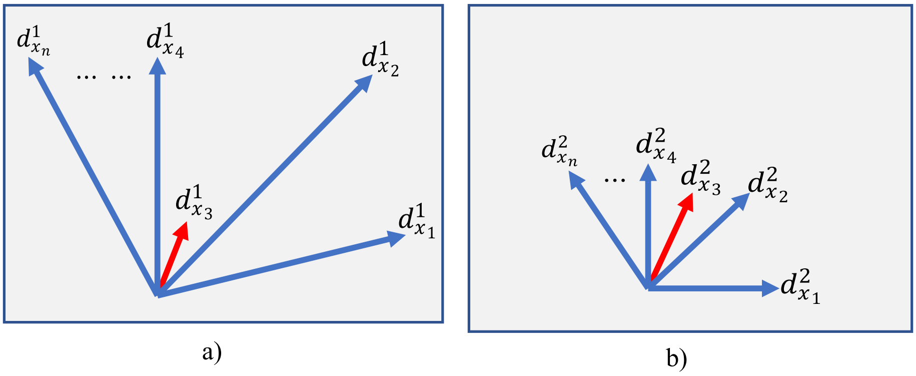


Fig. 2 Euclidean distances of *classifier*^{1} (**a**) and *classifier*^{2} (**b**) for a test signal

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 classifier's performance. For example, 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 a test signal. Based on this idea, the performance score was used for classification with the proposed MPSA. By the MPSA, the classification is made according to the obtained performance score values instead of the Euclidean distance criterion. The classifier that gives the highest performance of the two classifiers is determined as the base classifier. 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, a test signal is projected into the optimum subspaces separately for two classifiers. Finally, the Euclidean distances obtained for n classes are shown as vectors according to their magnitudes.

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

$$\mathbf{SC}^p_{ij} = \left(\frac{d^p_{x_{min}}}{d^p_{x_{ij}}}\right)^2 + \left(\frac{d^p_{x_{min}}}{d^p_{x_{ij}}}\right)^2 \ldots + \left(\frac{d^p_{x_{min}}}{d^p_{x_{ij}}}\right)^2, \quad i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, k, \tag{8}$$

where $d^p_{x_{min}}$ 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 signals, and k is the number of samples. The performance values of the classifiers are obtained using the MPSA 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^2_{ij}, SC^1_{ij}). Then the differences between the performance score values for the *jth* test signal of the *ith* class are found ($diff_{i,j} = SC^2_{ij}$-SC^1_{ij}). If this difference is positive, *classifier*^{1} has better performance, and the difference is assigned to the performance value ($perf^1$) of the classifier^1. If the difference is negative, *classifier*^{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 as the base classifier; otherwise, the second classifier is chosen as the base classifier. 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
    $diff_{i,j} = SC_{ij}^2\text{-}SC_{ij}^1$
      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
$ts^1 = summation(perf^1)$
$ts^2 = summation(perf^2)$
if $ts^1 > ts^2$
$b\_classifier = classifier^1$
else
$b\_classifier = classifier^2$
end for
end algorithm
```

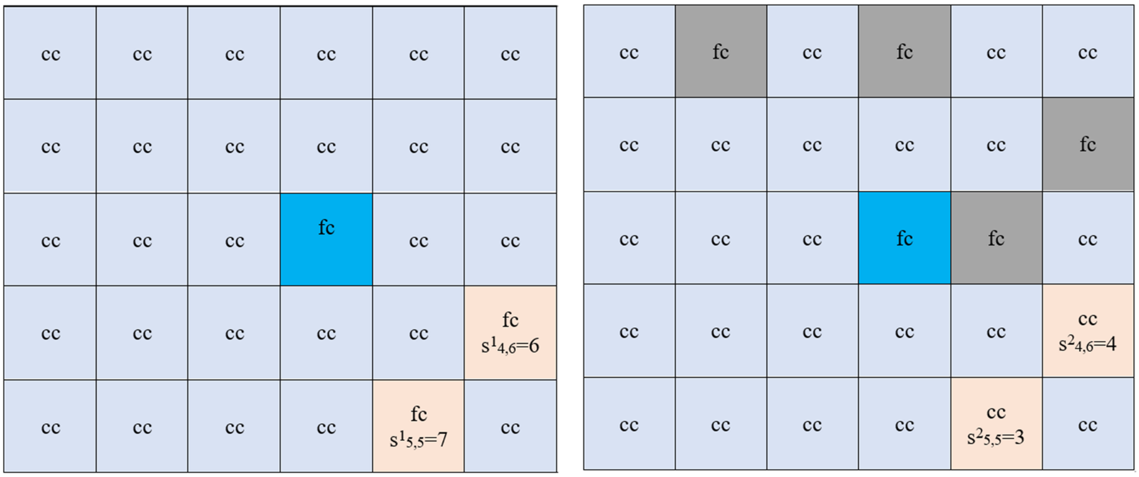
The base classifier is selected using the MPSA, and then the RUA is applied. The *euc_class*^{1} and *euc_class*^{2} matrices mentioned above are converted to *class*^{1} and *class*^{2} matrices containing the values cc when classified correctly and *fc* when classified incorrectly for each test signal.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

2.3.2 Recognition update algorithm (RUA)

The RUA is based on the base classifier's recognition matrix (*euc_class*) obtained according to the Euclidean values. Misclassified parts in this matrix are updated according to the performance score matrices of the base and another 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; otherwise, the updated recognition matrix receives the value of "*fc* (false classification)". The *updated_rec* matrix is size $n \times k$ 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
```



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 (*class*^{1})

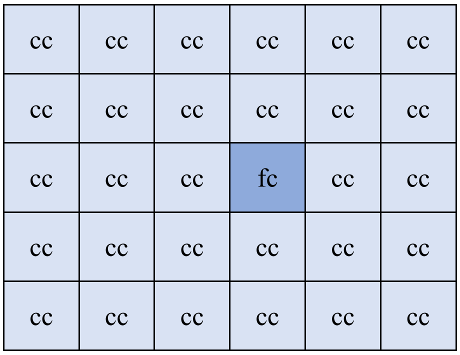
cc	fc	cc	fc	cc	cc
cc	cc	cc	cc	cc	fc
cc	cc	cc	fc	fc	cc
cc	cc	cc	cc	cc	cc $s^2_{4,6}=4$
cc	cc	cc	cc	cc $s^2_{5,5}=3$	cc

b) Other classifier matrix (*class*^{2})

Fig. 3 Base classifier (**a**) and another classifier (**b**) matrices

To indicate the RUA more clearly, the recognition matrices of the two classifiers (*class*^{1} and *class*^{2}) are given in Fig. 3. The first classifier (*classifier*^{1}) is assumed to be selected as the base classifier using the MPS algorithm. In this case, only the recognition matrix (*class*^{1}) of *classifier*^{1} is referenced.

As you can see, there are three errors in the *class*^{1} matrix. First, blue boxes in Fig. 3a, 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.



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 Updated recognition matrix

2.4 The proposed CNN classifier

The proposed CNN model consists of three convolution layers, 3 max-pooling layers, and 4 regularisation layers. A total of 50 epochs with 30 iterations were applied, and 0.01 was chosen as the learning coefficient. The adaptive moment estimation (Adam) optimiser was used as a solver for the training network. The layers of the proposed CNN model are shown in Fig. 5.

2.5 Transfer learning Alexnet model (TrAlexnet)

While CNN can give good results in face recognition for large databases [30], the Alexnet pre-trained CNN model can provide better results for small databases than classical CNN [31]. Therefore, the pre-trained Alexnet CNN model was also used in the study. All images are resized to 227 × 227, and all greyscale images are converted to RGB to be used in Alexnet. In the MATLAB environment, Alexnet consists of 25 layers, and the last three layers (fully connected, softmax, classification layers) are used to classify the features obtained from the previous layers. In transform learning, the layers are transferred to the new classification task by removing the last three layers and adding a new fully connected layer according to the number of classes in the database. So, a new fine-tuning deep transfer learning model for the problem is created. For this model, we used a 0.0001 learning rate, stochastic gradient descent with momentum (SGDM) optimiser, and 20 epochs.

2.6 The proposed Alexnet + SVM and Alexnet + KNN

Another deep learning model has been proposed apart from the TrAlexnet model used in the study. This model obtains some layers of Alexnet for feature extraction. These features

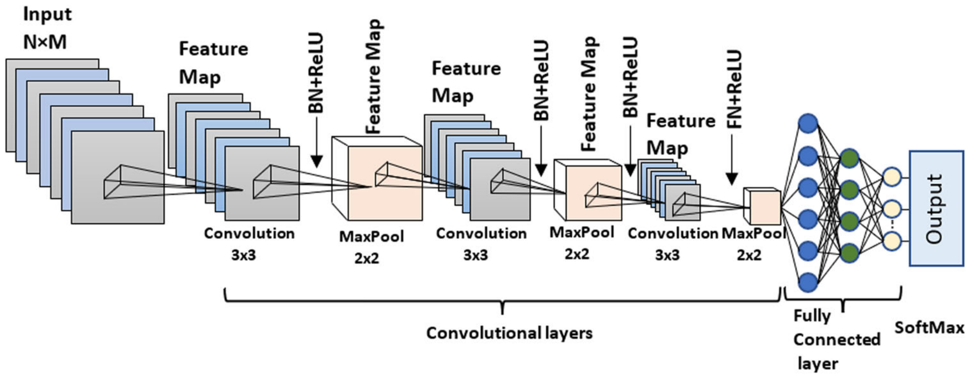


Fig. 5 Layers of the proposed CNN model

are used for classification using machine learning algorithms such as KNN and SVM [31, 32]. In this model, during the training phase, the 20th layer of the Alexnet model, fully connected-7 (fc7), was used to extract the feature. This way, 4096-dimensional features were obtained for each image and used in training for the SVM and KNN classifiers. The test images were classified using SVM and KNN classifiers in the test phase. Here, the linear kernel for SVM and K = 5 nearest neighbor is used for KNN. Besides, hyper-parameters proposed for TrAlexnet are used.

3 Experimental results

ORL, Extended YALE-B, YALE, and FRLL 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. In the experiments, 150 images were used because two images of each person were similar. The Extended Yale B database, which is the cropped version, contains 2414 images with a size of 192 × 168 over 38 subjects and 64 images per subject [29]. The extended Yale Face Database's 44 images were used for training and 20 for testing. In this way, threefold cross-validation was performed in the study. The FRLL database contains 1020 images with a size of 1350 × 1350 over 102 subjects and 10 images per subject [33]. Using a MATLAB code, each image in the FRLL database is resized by selecting the face region of 800 × 800.

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, 80 × 80, 40 × 40, 20 × 20, and 10 × 10 sized images were used for YALE. Also, 192 × 168, 96 × 84, 48 × 42 and , 24 × 21 sized images were used for the Extended YALE-B and, 160 × 160, 80 × 80, 40 × 40 , and 20 × 20 sized images were obtained for the FRLL database. For the deep learning methods using Alexnet, the sizes of the images were first reduced to the dimensions mentioned above. Then, the sizes of the images were resized to 227 × 227 pixels through MATLAB's augmented image datastore function. Experimental results were obtained using codes written in a MATLAB environment. The results for ORL and YALE databases are given in Tables 1 and 2, respectively. The + and – symbols next to the expressions showing the image dimensions indicate the sufficient and insufficient data cases, respectively.

Table 1 Recognition rates found for the ORL database

Classifiers	Image sizes			
	$64 \times 64^{-}$	$32 \times 32^{-}$	$16 \times 16^{+}$	$8 \times 8^{+}$
DCVA	99.50	99.25	99.00	97.75
Fisherface	99.00	99.00	97.00	99.25
HDF	99.75	99.50	99.50	99.50
CNN	96.75	97.75	93.50	91.25
Alexnet + Svm	97.50	95.00	92.50	82.50
Alexnet + Knn	97.50	96.75	93.00	83.00
TrAlexnet	98.50	97.75	94.50	89.50

Table 2 Recognition rates found for the YALE database

Classifiers	Image sizes			
	$80 \times 80^{-}$	$40 \times 40^{-}$	$20 \times 20^{-}$	$10 \times 10^{+}$
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
CNN	94.00	93.34	93.34	92.00
Alexnet + Svm	92.33	87.67	87.34	83.67
Alexnet + Knn	92.83	90.67	86.67	85.00
TrAlexnet	95.00	93.67	92.67	90.67

Table 3 Recognition rates found for the Extended YALE B database

Classifiers	Image sizes			
	$192 \times 168^{-}$	$96 \times 84^{-}$	$48 \times 42^{-}$	$24 \times 21^{+}$
DCVA	98.28	96.57	96.05	93.15
Fisherface	99.07	97.36	97.10	94.86
HDF	99.21	98.15	97.89	96.18
CNN	98.16	97.50	95.78	94.86
Alexnet + Svm	94.11	93.13	88.23	76.47
Alexnet + Knn	97.57	97.04	92.31	83.75
TrAlexnet	98.03	97.24	96.38	94.14

Table 4 Recognition rates found for the FRLL database

Classifiers	Image sizes			
	$160 \times 160^{-}$	$80 \times 80^{-}$	$40 \times 40^{-}$	$20 \times 20^{+}$
DCVA	58.28	56.07	54.95	47.05
Fisherface	70.49	69.11	70.58	70.49
HDF	78.43	77.25	77.64	76.47
CNN	87.59	88.33	86.27	85.29
Alexnet + Svm	91.18	88.24	78.43	68.82
Alexnet + Knn	98.04	94.12	83.34	71.76
TrAlexnet	96.08	94.21	92.12	88.23

Tables 1 and 2 show that the Hybrid-DF classifier has higher recognition rates than the Fisherface and DCVA classifiers. In Table 2, the highest difference between the recognition rates of the HDF and others was 2%, which was found using 40 × 40 images for the YALE database, and for the ORL database, the highest difference is 0.5% in Table 1. As seen in Table 1, CNN has a lower recognition rate than other classifiers.

These tables and the tables below show that DCVA has low recognition rates for sufficient data cases. The main reason is that the difference and indifference subspaces cannot be distinguished precisely. In these tables, TrAlexnet gave the best recognition results among deep learning methods, but HDF has higher recognition rates than all classifiers. The results obtained using the Extended YALE B database are given in Table 3.

In Table 3, the CNN gave higher recognition rates than the DCVA and Fisherface at some image sizes; however, the HDF has the highest recognition rate for the Extended YALE B database. While CNN and TrAlexnet obtained high recognition performance for sufficient and insufficient data cases, Alexnet + SVM and Alexnet + KNN gave lower recognition performance than CNN and TrAlexnet. The recognition results of the classifiers for the FRLL database are given in Table 4. One of the reasons for choosing the FRLL database is that the number of classes is higher than other databases. As it is known, as the number of classes increases, the projections of the images of each class into the optimum subspace will become more interfere and negatively affect the recognition rate. Finally, the FRLL database was chosen to examine the effects of this situation on subspace classifiers.

As can be seen from Table 4, deep learning algorithms have higher recognition rates than subspace classifiers for insufficient data cases. For insufficient data case, while TrAlexnet CNN, and Alexnet + KNN gave better recognition performance than other classifiers, CNN and TrAlexnet obtained the best recognition performances for sufficient data case. In subspace classifiers, HDF gave the best recognition performance for sufficient and insufficient data cases, which has approximately 8% higher recognition rates than Fisherface and 20% higher than DCVA. Moreover, DCVA achieved the lowest recognition rates for sufficient and insufficient data cases. In addition, HDF gave better recognition results than Alexnet + SVM and Alexnet + KNN for sufficient data cases.

In Table 5, the selected base classifiers using the MPSA algorithm are given for all dimensions of the images.

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. When all the experimental results are examined, subspace classifiers can exceed the classification performance of deep learning methods for databases with few classes. However, in the FRLL database, where the number of classes is higher than in other databases, it has been observed that subspace classifiers give lower recognition performance than deep learning algorithms.

4 Conclusions

Hybrid classifiers have an essential role in pattern recognition. Using heterogeneous classifiers in hybrid form can give significantly 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. Then the recognition matrix was updated using the RUA. The experimental studies showed that the MPSA correctly chose the base classifier corresponding to the classifier with the highest recognition

Table 5 Selected base classifiers using the MPSA

Databases							
YALE		ORL		Extended YALE B		FRLL	
Image sizes	Chosen Classifiers	Image sizes	Chosen Classifiers	Image sizes	Chosen Classifiers	Image sizes	Chosen Classifiers
80 × 80	Fisherface	64 × 64	DCVA	192 × 168	Fisherface	160 × 160	Fisherface
40 × 40	DCVA	32 × 32	DCVA	96 × 84	Fisherface	80 × 80	Fisherface
20 × 20	Fisherface	16 × 16	DCVA	48 × 42	Fisherface	40 × 40	Fisherface
10 × 10	Fisherface	8 × 8	Fisherface	24 × 21	Fisherface	20 × 20	Fisherface

rate for all test signals and image sizes. It was observed in experimental studies that the proposed hybrid classifier gives higher recognition rates than DCVA and Fisherface for all image dimensions. In addition, the recognition performances of Alexnet + SVM, Alexnet + KNN, and TrAlexnet deep learning algorithms and subspace classifiers were compared. As a result of the comparison, subspace classifiers generally have better recognition performance for databases such as YALE, ORL, and Extended YALE B with a small number of classes. At the same time, deep learning algorithms perform better recognition performance than subspace classifiers in sufficient and insufficient data cases for databases with more classes, such as the FRLL database. This result obtained by the subspace classifiers is related to more interference between the class signals projected to the optimum subspace due to the increased number of classes. The proposed HDF subspace classifier has better recognition performance than DCVA and Fisherface in all experimental studies. Finally, the results show that HDF is a classifier with high recognition performance, especially for datasets with small classes.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11760-022-02468-w.

Author contributions All the work done to create this article was carried out only by Keser.

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

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

Declarations

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

References

1. Woźniak, M., Grana, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fusion **16**, 3–17 (2014). https://doi.org/10.1016/j.inffus.2013.04.006
2. Giacinto, G., Roli, R.: An approach to the automatic design of multiple classifier systems. Pattern Recogn. Lett. **22**, 25–33 (2001). https://doi.org/10.1016/S0167-8655(00)00096-9
3. Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. J. Netw. Comput. Appl. **28**(2), 167–182 (2005). https://doi.org/10.1016/j.jnca.2004.01.003
4. Gu S., Jin Y.: Heterogeneous classifier ensembles for EEG-based motor imaginary detection. In: 2012 12th UK Workshop on Computational Intelligence (UKCI), pp. 1–8. IEEE (2012). https://doi.org/10.1109/UKCI.2012.6335751
5. Cheng J., Chen L.: A weighted regional voting based ensemble of multiple classifiers for face recognition. In: International Symposium on Visual Computing, pp 482–91. Springer (2014). https://doi.org/10.1007/978-3-319-14364-4_46
6. Govindarajan M., Chandrasekaran R.: Intrusion detection using an ensemble of classification methods. In: World congress on engineering and computer science, vol. 1, pp. 1–6 (2012).
7. Nweke, H.F., Teh, Y.W., Mujtaba, G., Al-Garadi, M.A.: Data fusion and multiple classifier systems for human activity detection and health monitoring: review and open research directions. Inf. Fusion **46**, 147–170 (2019). https://doi.org/10.1016/j.inffus.2018.06.002
8. Mi, J.X., Huang, D.S., Wang, B., Zhu, X.: The nearest-farthest subspace classification for face recognition. Neurocomputing **113**, 241–250 (2013). https://doi.org/10.1016/j.neucom.2013.01.003
9. Li, P., Song, Y., Wang, P., Dai, L.: A multi-feature multi-classifier system for speech emotion recognition. In: 2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia), pp. 1–6. IEEE(2018).
10. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: a new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1619–1630 (2006). https://doi.org/10.1109/TPAMI.2006.211
11. Liao, M., Gu, X.: Face recognition approach by subspace extended sparse representation and discriminative feature learning. Neurocomputing **373**, 35–49 (2020). https://doi.org/10.1016/j.neucom.2019.09.025
12. Liao, M., Gu, X.: Subspace clustering based on alignment and graph embedding. Knowl.-Based Syst. **188**, 105029 (2020). https://doi.org/10.1016/j.knosys.2019.105029
13. Liao, M., Li, Y., Gao, M.: Graph-based adaptive and discriminative subspace learning for face image clustering. Expert Syst. Appl. **192**, 116359 (2022). https://doi.org/10.1016/j.eswa.2021.116359
14. Xu, Y., Zhang, D., Yang, J., Yang, J.Y.: A two-phase test sample sparse representation method for use with face recognition. IEEE

Trans. Circuits Syst. Video Technol. **21**(9), 1255–1262 (2011). https://doi.org/10.1109/TCSVT.2011.2138790
15. Li, B., Zheng, C.H., Huang, D.S.: Locally linear discriminant embedding: an efficient method for face recognition. Pattern Recogn. **41**(12), 3813–3821 (2008). https://doi.org/10.1016/j.patcog.2008.05.027
16. Fan, Z., Xu, Y., Zhang, D.: Local linear discriminant analysis framework using sample neighbors. IEEE Trans. Neural Netw. **22**(7), 1119–1132 (2011). https://doi.org/10.1109/TNN.2011.2152852
17. Pang, Y., Yuan, Y., Wang, K.: Learning optimal spatial filters by discriminant analysis for brain–computer-interface. Neurocomputing **77**(1), 20–27 (2012). https://doi.org/10.1016/j.neucom.2011.07.016
18. Huang, D.S., Du, J.X.: A constructive hybrid structure optimisation methodology for radial basis probabilistic neural networks. IEEE Trans. Neural Netw. **19**(12), 2099–2115 (2008). https://doi.org/10.1109/TNN.2008.2004370
19. Ebied, H. M.: Feature extraction using PCA and Kernel-PCA for face recognition. In: 2012 8th International Conference on Informatics and Systems (INFOS), pp MM-72. IEEE (2012).
20. Hidayat, E., Fajrian, N. A., Muda, A. K., Huoy, C. Y., Ahmad, S.: 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 (2011).
21. Koç, M., Barkana, A.: A new solution to one sample problem in face recognition using FLDA. Appl. Math. Comput. **217**(24), 10368–10376 (2011). https://doi.org/10.1016/j.amc.2011.05.048
22. Sun, Z., Li, J., Sun, C.: Kernel inverse Fisher discriminant analysis for face recognition. Neurocomputing **134**, 46–52 (2014). https://doi.org/10.1016/j.neucom.2012.12.075
23. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997). https://doi.org/10.1109/34.598228
24. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **27**(1), 4–13 (2005). https://doi.org/10.1109/TPAMI.2005.9
25. He, Y. H., Zhao, L., Zou, C. R.: Kernel discriminative common vectors for face recognition. In: 2005 International Conference on Machine Learning and Cybernetics,vol. 8, pp. 4605–4610. IEEE (2005).
26. Wen, Y.: An improved discriminative common vectors and support vector machine based face recognition approach. Expert Syst. Appl. **39**(4), 4628–4632 (2012). https://doi.org/10.1016/j.eswa.2011.09.119
27. Zhu, L., Jiang, Y., Li, L.: 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 (2008). https://doi.org/10.1109/ICCIS.2008.4670909
28. Wen, Y.: A novel dictionary based SRC for face recognition. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2582–2586. IEEE (2017).
29. Lee, K.C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005). https://doi.org/10.1109/TPAMI.2005.92
30. Parkhi, O. M., Vedaldi, A., Zisserman, A.: Deep face recognition (2015).
31. Almabdy, S., Elrefaei, L.: Deep convolutional neural network-based approaches for face recognition. Appl. Sci. **9**(20), 4397 (2019). https://doi.org/10.3390/app9204397
32. Khan, S., Ahmed, E., Javed, M. H., Shah, S. A., Ali, S. U.: Transfer learning of a neural network using deep learning to perform face recognition. In: 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp 1–5. IEEE (2019).
33. DeBruine, L.M., Jones, B.C.: Face Research Lab London Set (2017).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.